

PAT-NO: JP02001005704A

DOCUMENT-IDENTIFIER: JP 2001005704 A

TITLE: SYSTEM AND METHOD FOR DATABASE ACCESS AND STORAGE MEDIUM

PUBN-DATE: January 12, 2001

INVENTOR-INFORMATION:

NAME COUNTRY
KODERA, SUNAO N/A

ASSIGNEE-INFORMATION:

NAME COUNTRY
NEC SOFTWARE CHUGOKU LTD N/A

APPL-NO: JP11172028

APPL-DATE: June 18, 1999

INT-CL (IPC): G06F012/00, G06F017/30

ABSTRACT:

PROBLEM TO BE SOLVED: To increase the efficiency of data access and to simplify a program by generating a data table of unique keys permitting direct access to data that an application requires on a database.

SOLUTION: A database management system 5 controls and accesses the database 4 stored with data. A stored procedure executing means 6 executes a stored procedure registered in the database management system 5 and generates the data table of unique keys defined in the stored procedure. A database access request means 7 requests the database management system 5 to access the database 4 by using the data table of unique keys as the execution result of the stored procedure executing means 6 and receives the request result from the database management system 5.

COPYRIGHT: (C)2001JPO

*translation request
Submitted*

Best Available Copy

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2001-5704
(P2001-5704A)

(43)公開日 平成13年1月12日(2001.1.12)

(51)Int.Cl. ⁷	識別記号	F I	テームト* (参考)
G 0 6 F 12/00	5 1 3	G 0 6 F 12/00	5 1 3 J 5 B 0 7 5
17/30		15/40	3 1 0 F 5 B 0 8 2
			3 8 0 D
		15/403	3 2 0 Z

審査請求 有 請求項の数7 OL (全 11 頁)

(21)出願番号 特願平11-172028

(22)出願日 平成11年6月18日(1999.6.18)

(71)出願人 000211329

中国日本電気ソフトウェア株式会社
広島県広島市南区稲荷町4番1号

(72)発明者 小寺 直

広島県広島市南区稲荷町4番1号 中国日
本電気ソフトウェア株式会社内

(74)代理人 100082935

弁理士 京本 直樹 (外2名)

Fターム(参考) 5B075 NK02 NK10 NK24 NK54 PQ02

PQ05

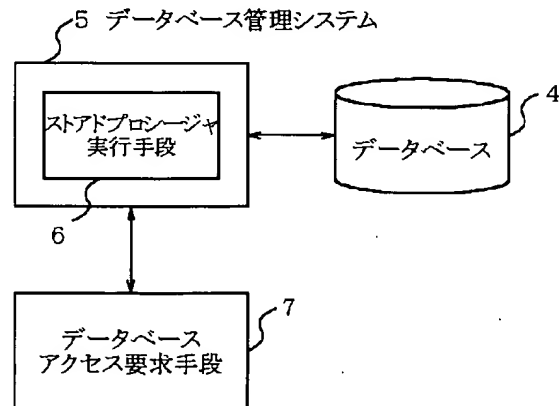
5B082 BA09 EA05 GA08

(54)【発明の名称】 データベースアクセス方式、方法および記憶媒体

(57)【要約】

【課題】 データベース上で、アプリケーションが必要とするデータへの直接アクセスが可能な一意キーのデータテーブルを作成することで、データアクセスの効率アップとプログラムの簡略化を実現する。

【解決手段】 データベース管理システム5は、データを記憶蓄積するデータベース4を管理しアクセスする。ストアードプロシージャ実行手段6は、データベース管理システム5に登録されたストアードプロシージャを実行し、ストアードプロシージャに定義された一意キーのデータテーブルを作成する。データベースアクセス要求手段7は、ストアードプロシージャ実行手段6の実行結果である一意キーのデータテーブルを利用してデータベース4にアクセスすることをデータベース管理システム5に要求し要求結果をデータベース管理システム5から受け取る。



【特許請求の範囲】

【請求項1】 データを記憶蓄積するデータベースと、データベースアクセス要求手段からの要求を受け付け前記データベースにアクセスしアクセス結果を前記データベースアクセス要求手段に返すデータベース管理システムと、前記データベース管理システムに含まれストアードプロシーダを実行しストアードプロシーダに定義された一意キーの表を作成するストアードプロシーダ実行手段と、前記ストアードプロシーダ実行手段が作成した一意キーの表を参照して前記データベースにアクセスする要求を前記データベース管理システムに行うデータベースアクセス要求手段と、を備えることを特徴とするデータベースアクセス方式。

【請求項2】 サーバとクライアントがネットワークを介して接続されるクライアントサーバシステムにおいて、前記サーバは、データを記憶蓄積し1つ以上の実表から成る関係データベースと、アプリケーションから要求された処理に基づき前記関係データベースにアクセスし処理結果を前記アプリケーションに返す関係データベース管理システムと、前記実表と関係づけた仮想表を定義し一意キーの表を作成するストアードプロシーダと、前記ストアードプロシーダを起動するストアードプロシーダ起動手段とを備え、前記クライアントは、前記ストアードプロシーダが定義する仮想表を指定して前記関係データベースにアクセスする要求を前記関係データベース管理システムに発行し要求結果を前記関係データベース管理システムから受け取るアプリケーションを備えることを特徴とするデータベースアクセス方式。

【請求項3】 前記ストアードプロシーダ起動手段は、前記ストアードプロシーダが既に起動実行され且つその後には前記関係データベースの更新がなされていない場合には前記ストアードプロシーダを起動しないことを特徴とする請求項2記載のデータベースアクセス方式。

【請求項4】 前記ストアードプロシーダが定義する仮想表は、アプリケーションが前記関係データベースにアクセスするときにインデックスの役目をすることを特徴とする請求項2記載のデータベースアクセス方式。

【請求項5】 サーバとクライアントがネットワークを介して接続されるクライアントサーバシステムにおいてデータベースにアクセスする方法であって、アプリケーションは実表とストアードプロシーダで定義されている仮想表とを関係づけた表を定義して関係データベースに対するアクセス要求を関係データベース管理システムに行い、前記関係データベース管理システムは前記アプリケーションからのアクセス要求を分析し、前記ストアードプロシーダが定義生成する仮想表が指定されているときはストアードプロシーダを起動実行してストアードプロシーダに定義された仮想表を生成し、前記仮想表を使用したアプリケーションのアクセス要求を処理してアプリケーションに処理結果を返し、アプリケーションは

関係データベース管理システムから処理結果を受け取ることを特徴とするデータベースアクセス方法。

【請求項6】 前記ストアードプロシーダは基本データの実表とその履歴を保持する2つ以上の実表とから最新の履歴の一意キーから成る仮想表を生成する処理であることを特徴とする請求項5記載のデータベースアクセス方法。

【請求項7】 アプリケーションが実表とストアードプロシーダで定義されている仮想表とを関係づけた表を定義して関係データベースに対するアクセス要求を関係データベース管理システムに行うアクセス要求処理と、前記関係データベース管理システムが前記アプリケーションからのアクセス要求を分析し、前記ストアードプロシーダが定義生成する仮想表が指定されているときは前記ストアードプロシーダを起動するストアードプロシーダ起動処理と、前記ストアードプロシーダを実行してストアードプロシーダに定義された仮想表を生成する仮想表生成処理と、前記仮想表を使用したアプリケーションのアクセス要求を処理してアプリケーションに処理結果を返すアクセス処理と、アプリケーションが前記関係データベース管理システムから処理結果を受け取るデータ受取処理と、をコンピュータに実行させるためのプログラムを記録したことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データベース管理システムを用いてデータベースにアクセスするデータベースアクセス方式、方法および記憶媒体に関し、特にデータベース管理システムに登録したストアードプロシーダを用いてデータベースにアクセスするデータベースアクセス方式、方法および記憶媒体に関する。

【0002】

【従来の技術】データベース管理システムの一つである関係データベース管理システム（以下、RDBMSと略称する）では、関係データベース（以下、RDBと略称する）を表形式で管理している。表には、実際にデータが格納されている実表とデータは保存されず実行時に元になる実表のデータから生成される仮想表（ビューとも称す）がある。アプリケーションは表を参照するSQL言語を用いてRDBMSを介してRDB上の必要なデータを取得する。

【0003】クライアント／サーバシステム（以下、C/Sシステムと略称する）においては、サーバにRDBを接続してRDBMSを備え、クライアント側に存在するアプリケーションがSQL言語を用いてRDBMSに対してRDBアクセス要求を行い、RDBMSからアクセス結果を貰う形態になっている。

【0004】すなわち、クライアントのアプリケーションが必要なデータを検索する場合、アプリケーションはSQL言語の書式にあわせた検索条件式をサーバのRDB

BMSに発行し、RDBMSがRDBのデータを検索して与えられた検索条件に合致するデータを選択し、クライアントのアプリケーションに検索結果を返す。このとき、RDBMSにおいて、検索用インデックスの作成などの高速アクセス技法が使用される。

【0005】しかし、RDBMSで作成されるインデックスがアプリケーションに最適な形で作成されるとは限らない。

【0006】例えば、図10(a)、(b)に示すような実表Aと実表Bから成るRDBにおいて、商品ごとに最新の値段を求めるアプリケーションについて考えてみる。

【0007】アプリケーションは、以下の処理を行う。

(1) 実表Bにおいて、同一商品番号の中で日付時刻が最新のものを含む行を選択する。

(2) 実表Aと上記(1)の結果を、それぞれの商品番号で関係づける。

(3) 上記(2)の関係づけにより、商品番号と商品名称と値段から成る表を生成する。

(4) 上記(3)の結果である表にアクセスして、商品番号と商品名称と値段のデータを取得する。

【0008】この場合、RDBMSにおいて日付時刻のインデックスは作成されるが、同一商品番号の中で日付時刻が最新のものに対するインデックスは作成されない。このため、アプリケーションは同一商品番号の中で日付時刻が最新のものを選択するための処理手続を、上記(1)において記述しなければならない。このように、RDBMSが作成するインデックスは個々のアプリケーションに最適なものではない。

【0009】また、上記のアプリケーション処理は各クライアントで実行されるので、各クライアントからの要求情報とサーバからの結果情報が、ネットワークを介してサーバとクライアント間で頻繁に行き交うことになる。

【0010】

【発明が解決しようとする課題】上述した従来の技術では、RDBMSで作成されるインデックスはデータ項目の単純なインデックスであり、アプリケーション側で特に必要とするデータがインデックス形式にならずアプリケーションの作成が煩雑になるという問題点がある。また、そのために、RDBMSとアプリケーションとの間のトラフィックが高まるという問題点もある。

【0011】本発明の目的は、RDBMSに登録されたストアドプロシージャを用いて、アプリケーションごとに必要なインデックスを作成し、アプリケーション作成を簡易化し、システム全体の性能を向上させる手段を提供することにある。

【0012】

【課題を解決するための手段】本願第1の発明のデータベースアクセス方式は、データを記憶蓄積するデータ

ベースと、データベースアクセス要求手段からの要求を受け付け前記データベースにアクセスしアクセス結果を前記データベースアクセス要求手段に返すデータベース管理システムと、前記データベース管理システムに含まれストアドプロシージャを実行しストアドプロシージャに定義された一意キーの表を作成するストアドプロシージャ実行手段と、前記ストアドプロシージャ実行手段が作成した一意キーの表を参照して前記データベースにアクセスする要求を前記データベース管理システムに行うデータベースアクセス要求手段と、を備える。

【0013】本願第2の発明のデータベースアクセス方式は、サーバとクライアントがネットワークを介して接続されるクライアントサーバシステムにおいて、前記サーバは、データを記憶蓄積し1つ以上の実表から成る関係データベースと、アプリケーションから要求された処理に基づき前記関係データベースをアクセスし処理結果を前記アプリケーションに返す関係データベース管理システムと、前記実表と関係づけた仮想表を定義し一意キーの表を作成するストアドプロシージャと、前記ストアドプロシージャを起動するストアドプロシージャ起動手段とを備え、前記クライアントは、前記ストアドプロシージャが定義する仮想表を指定して前記関係データベースにアクセスする要求を前記関係データベース管理システムに発行し要求結果を前記関係データベース管理システムから受け取るアプリケーションを備える。

【0014】本願第3の発明のデータベースアクセス方式は、第2の発明において前記ストアドプロシージャ起動手段は、前記ストアドプロシージャが既に起動実行され且つその後前記関係データベースの更新がなされていない場合には前記ストアドプロシージャを起動しないことを特徴とする。

【0015】本願第4の発明のデータベースアクセス方式は、第2の発明において前記ストアドプロシージャが定義する仮想表は、アプリケーションが前記関係データベースをアクセスするときにインデックスの役目をすることを特徴とする。

【0016】本願第5の発明のデータベースアクセス方法は、サーバとクライアントがネットワークを介して接続されるクライアントサーバシステムにおいてデータベースにアクセスする方法であって、アプリケーションは実表とストアドプロシージャで定義されている仮想表とを関係づけた表を定義して関係データベースに対するアクセス要求を関係データベース管理システムに行い、前記関係データベース管理システムは前記アプリケーションからのアクセス要求を分析し、前記ストアドプロシージャが定義生成する仮想表が指定されているときはストアドプロシージャを起動実行してストアドプロシージャに定義された仮想表を生成し、前記仮想表を使用したアプリケーションのアクセス要求を処理してアプリケーションに処理結果を返し、アプリケーションは関係

5

データベース管理システムから処理結果を受け取ることを特徴とする。

【0017】本願第6の発明のデータベースアクセス方法は、第5の発明において前記ストアプロシージャは基本データの実表とその履歴を保持する2つ以上の実表とから最新の履歴の一意キーから成る仮想表を生成する処理であることを特徴とする。

【0018】本願第7の発明の記録媒体は、アプリケーションが実表とストアプロシージャで定義されている仮想表とを関係づけた表を定義して関係データベースに対するアクセス要求を関係データベース管理システムに行うアクセス要求処理と、前記関係データベース管理システムが前記アプリケーションからのアクセス要求を分析し、前記ストアプロシージャが定義生成する仮想表が指定されているときは前記ストアプロシージャを起動するストアプロシージャ起動処理と、前記ストアプロシージャを実行してストアプロシージャに定義された仮想表を生成する仮想表生成処理と、前記仮想表を使用したアプリケーションのアクセス要求を処理してアプリケーションに処理結果を返すアクセス処理と、アプリケーションが前記関係データベース管理システムから処理結果を受け取るデータ受取処理と、をコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【0019】

【発明の実施の形態】本発明のデータベースアクセス方式について、図1を参照して説明する。

【0020】図1を参照すると、本発明のデータベースアクセス方式は、データベース4とデータベース管理システム5とストアプロシージャ実行手段6とデータベースアクセス要求手段7から構成されている。

【0021】データベース4は、データを記憶蓄積するデータベースである。

【0022】データベース管理システム5は、データベース4を管理し、データベースアクセス要求手段7からの要求に基づきデータベース4にアクセスしてアクセス結果をデータベースアクセス要求手段7に返す。

【0023】ストアプロシージャ実行手段6は、データベース管理システム5に登録されたストアプロシージャを実行する。ストアプロシージャは、データベースアクセス要求手段7が使用する一意キーの表（インデックス）を作成する処理を含んでいる。

【0024】データベースアクセス要求手段7は、ストアプロシージャ実行手段6を実行させ実行結果を利用してデータベース4にアクセスすることをデータベース管理システム5に要求し要求結果をデータベース管理システム5から受け取る。

【0025】本発明の動作について説明する。なお、データベースアクセス要求手段7が利用するストアプロシージャは、あらかじめ公知の方法で、データベース管

6

理システム5に登録されているものとする。

【0026】まず、データベースアクセス要求手段7はストアプロシージャが作成する一意キーの表（インデックス）を指定してデータベース4に対するアクセス要求をデータベース管理システム5に送る。

【0027】データベース管理システム5はストアプロシージャ実行手段6を用いて指定されたストアプロシージャを実行してデータベースアクセス要求手段7が使用する一意キーの表（インデックス）を作成した後に指定されたアクセス要求処理を行い、処理結果をデータベースアクセス要求手段7に返す。

【0028】データベースアクセス要求手段7はデータベース管理システム5から処理結果を受け取る。

【0029】このように、データベース管理システム5に登録されたストアプロシージャにおいてデータベースアクセス要求手段7が使用する一意キーの表（インデックス）を作成することにより、データベースアクセス要求手段7は求めるデータを高速にデータベース4から取得することができる。

【0030】本発明のデータベースアクセス方式を適用した本発明の実施の形態について説明する。

【0031】本発明の第1の実施の形態について、図面を参照して詳細に説明する。第1の実施の形態は、2つの実表から成る商品の最新値段を管理しているRDBを検索して商品の最新値段を取得して処理するシステムに関するものである。

【0032】図2は、第1の実施の形態の構成を示す図である。

【0033】図2を参照すると、第1の実施の形態は、サーバ1とクライアント2がネットワーク3で接続されている。サーバ1は、RDB11を接続し、RDBMS12を備えている。RDBMS12はストアプロシージャ13とストアプロシージャ起動手段14とを含む。クライアント2は、アプリケーション21を含む。

【0034】RDB11は、データを記憶蓄積する関係データベースで、実表A111と実表B112とから成っている。実表A111は基本データであり、実表B112は基本データの履歴を保持するものである。

【0035】実表A111は、RDB11を構成している表の一つである。実表A111の構成を図3(a)に示す。実表A111は商品番号と商品名称とから成り、各行は主キーである商品番号により一意に区別される。

【0036】実表B112は、RDB11を構成している表の一つである。実表B112の構成を図3(b)に示す。実表B112は商品番号と日付時刻と値段とから成り、商品の最新の値段を示している。商品の値段が変更される度に、変更した日付時刻と新しい値段を示す行が登録される。商品番号は実表A111の商品番号と対応しており、これにより実表A111と実表B112が関係づけられる。日付時刻は値段が登録される日付と時

刻であり、同一商品番号内での値段の登録順番を示すものである。日付時刻に替えて一連番号でもよい。商品番号と日付時刻を連結したもの（以下、「商品番号+日付時刻」と記す）で各行を一意に区別する。

【0037】RDBMS12は、RDB11を管理しアクセスする。アプリケーション21から発行された要求を処理し、発行された要求に対応する処理結果をアプリケーション21に返す。

【0038】ストアプロシージャ13は、実表A111と実表B112を基に仮想表A131を定義生成してデータセットを選択作成する処理を行う。ストアプロシージャ13は、ストアプロシージャ起動手段14により起動される。仮想表A131の名前は、ストアプロシージャ13をRDBMS12に登録するときに決定される。アプリケーション21はこの名前を指定することで、ストアプロシージャ13が生成した仮想表A131を参照することができる。なお、ストアプロシージャ13をRDBMS12に登録する方法については、公知の方法による。

【0039】仮想表A131は、最新の履歴の一意キーから成る仮想表である。実表A111と実表B112を基にストアプロシージャ13が定義生成し、実表A111の商品番号とそれに対応する実表B112の商品番号内で最新（最大）の日付時刻とを組み合わせる選択条件が指定してある。仮想表A131の構成を図3(c)に示す。仮想表A131は商品番号と日付時刻とから成り、商品番号は実表A111の商品番号に、日付時刻は実表B112の同一商品番号内の最新（最大）の日付時刻に、それぞれ対応している。仮想表A131は、アプリケーション21から見れば、常にRDB11の最新状態を反映したインデックスの役目をしており、アプリケーション21から参照される。

【0040】ストアプロシージャ起動手段14は、アプリケーション21がストアプロシージャ13が生成する仮想表A131を使用する要求をRDBMS12に行った場合に、ストアプロシージャ13を起動する。指定されたストアプロシージャ13が既に起動実行されていて且つその後にRDB11の更新がなされていない場合には、ストアプロシージャ13を起動しない。この場合、既に実行されたストアプロシージャ13の処理結果が使用される。

【0041】アプリケーション21は、RDB11の最新データを取得するために、ストアプロシージャ13が定義生成する仮想表A131と関係づけを行った仮想表B211を定義してアクセス要求をRDBMS12に発行する。発行した要求に対応する処理結果がRDBMS12から返され、取得したデータを基にデータ処理を行う。

【0042】仮想表B211は、仮想表A131と実表A111と実表B112を基にアプリケーション21が

定義生成する仮想表であり、アプリケーション21が必要とするデータから成る。仮想表B211の構成を図3(d)に示す。仮想表B211は商品番号と商品名称と値段とから成り、商品番号は仮想表A131の商品番号に対応する実表A111の商品番号に、商品名称は仮想表A131の商品番号に対応する実表A111の商品名称に、値段は仮想表A131の「商品番号+日付時刻」に対応する実表B112の値段に、それぞれ対応している。

【0043】本発明の第1の実施の形態の動作について、図1～図5を参照して詳細に説明する。

【0044】図4は第1の実施の形態の動作を説明する図で、(a)は実表A111の具体例を、(b)は実表B112の具体例を、(c)は仮想表A131の具体例を、(d)は仮想表B211の具体例を、示したものである。(a)と(b)から、商品名称“あああ”の最新の値段は日付時刻“05261800”に設定された“190”であることがわかる。図5は第1の実施の形態の動作の流れを示す図である。

【0045】図5を参照すると、アプリケーション21は、実表A111の商品番号と仮想表A131の商品番号とを関係づけ、実表B112の「商品番号+日付時刻」と仮想表A131の「商品番号+日付時刻」とを関係づけ、商品番号と商品名称と値段とから成る仮想表B211を定義して、RDB11に対するアクセスをRDBMS12に要求する（ステップA11～A12）。

【0046】RDBMS12は、アプリケーション21からのRDB11に対するアクセス要求を分析し（ステップM11）、ストアプロシージャ13が定義生成する仮想表A131が指定されているときは、ストアプロシージャ13の起動実行が必要であるか否かを判断し、ストアプロシージャ13の起動実行が必要であることを認識したときにストアプロシージャ13を起動する（ステップM12～M14）。指定されたストアプロシージャ13が既に起動実行されていて且つその後にRDB11の更新がなされていない場合には、ストアプロシージャ13を起動しない。この場合、既に実行されたストアプロシージャ13の処理結果が使用される。

【0047】起動されたストアプロシージャ13は以下の処理を実行する。

(1) 実表B112の商品番号ごとに、同一商品番号内で日付時刻が最新のものを含む行を選択する（ステップS11）。

(2) 実表A111の商品番号と上記(1)の結果の商品番号とを関係づける（ステップS12）。

(3) 実表A111の商品番号と上記(1)の結果の日付時刻から成る仮想表A131のデータセットを作成する（ステップS13）。ストアプロシージャ13の出力結果を図4(c)に示す。

【0048】続いて、RDBMS12は仮想表B211に基づく処理を行う(ステップM15)。RDBMS12の処理結果を図4(d)に示す。

【0049】その後、RDBMS12は処理結果をアプリケーション21に返す(ステップM16)。

【0050】アプリケーション21は、RDBMS12から処理結果を受け取り、アプリケーション処理を行う(ステップA13)。

【0051】このようにして、アプリケーション21は、ストアプロシージャ13の出力結果である仮想表A131をインデックス代わりに使用することにより、求めているRDB11の最新のデータを高速に得ることができる。

【0052】次に、本発明の第2の実施の形態について、図面を参照して詳細に説明する。第2の実施の形態は、3つの実表から成る製品の障害内容を管理しているRDBを検索して製品の最新の障害内容を取得して処理するシステムに関するものである。

【0053】図6は、第2の実施の形態の構成を示す図である。

【0054】図6を参照すると、第2の実施の形態は、サーバ1とクライアント2がネットワーク3で接続され、サーバ1はRDB15を接続しストアプロシージャ17とストアプロシージャ起動手段14とを含むRDBMS16を備え、クライアント2はアプリケーション22を備えて構成され、第1の実施の形態のRDB11とRDBMS12とストアプロシージャ13とアプリケーション21とが、RDB15とRDBMS16とストアプロシージャ17とアプリケーション22とに置換されたものである。

【0055】ここでは、第1の実施の形態と異なるRDB15とRDBMS16とストアプロシージャ17とアプリケーション22とについて説明する。

【0056】RDB15は、データを記憶蓄積する関係データベースで、実表A151と実表B152と実表C153とから成っている。実表A151は基本データであり、実表B152と実表C153は基本データの履歴を保持するものである。

【0057】実表A151は、RDB15を構成している表の一つである。実表A151の構成を図7(a)に示す。実表A151は製品番号と製品名称とから成り、各行は主キーである製品番号により一意に区別される。

【0058】実表B152は、RDB15を構成している表の一つである。実表B152の構成を図7(b)に示す。実表B152は製品番号と枝番Aと受付番号と削除とから成る。製品番号は実表A151の製品番号と対応しており、これにより実表B152と実表A151が関係づけられる。枝番Aは同一製品番号内での受付番号の順番を示すもので、一連番号でも日付時刻でもよい。

「製品番号+枝番A」で各行を一意に区別する。受付番

号は製品を通して採番される一連番号であり、製品の障害が受け付けられるごとに受付番号が登録される。障害の内容は実表C153に登録される。同一製品に対して新たな障害受付があった場合には、枝番Aを上げて新たな受付番号を登録する。削除は、受付番号が一旦受け付けられた後に削除されたか否かを示すもので、受付番号が削除された場合に「DEL」と示される。

【0059】実表C153は、RDB15を構成している表の一つである。実表C153の構成を図7(c)に示す。実表C153は製品番号と枝番Aと枝番Bと障害内容とから成る。「製品番号+枝番A」は実表B152の「製品番号+枝番A」と対応しており、これにより実表C153と実表B152が関係づけられる。枝番Bは同一「製品番号+枝番A」内での障害内容の順番を示すもので、一連番号でも日付時刻でもよい。「製品番号+枝番A+枝番B」で各行を一意に区別する。障害内容には、受付番号に対応した障害内容が登録される。受け付けた受付番号についての障害内容が更新される場合には、枝番Bを上げて更新された障害内容を登録し別のデータとして履歴管理を行う。

【0060】RDBMS16は、RDB15を管理しアクセスする。アプリケーション22から発行された要求を処理し、発行された要求に対応する処理結果をアプリケーション22に返す。

【0061】ストアプロシージャ17は、実表A151と実表B152と実表C153とを基に仮想表A171を定義生成してデータセットを選択作成する処理を行う。ストアプロシージャ17は、ストアプロシージャ起動手段14により起動される。仮想表A171の名前は、ストアプロシージャ17をRDBMS16に登録するときに決定される。アプリケーション22はこの名前を指定することで、ストアプロシージャ17が生成した仮想表A171を参照することができる。なお、ストアプロシージャ17をRDBMS16に登録する方法については、公知の方法による。

【0062】仮想表A171は、最新の履歴の一意キーから成る仮想表である。実表A151と実表B152と実表C153とを基にストアプロシージャ17が定義生成し、実表A151の製品番号、実表B152の同一製品番号内で最大の枝番Aおよび実表C153の同一「製品番号+枝番A」内で最大の枝番Bとを組み合わせた選択条件を指定してある。仮想表A171の構成を図7(c)に示す。仮想表A171は製品番号と枝番Aと枝番Bとから成り、製品番号は実表A151の製品番号に、枝番Aは実表B152の同一製品番号内で最大の枝番Aに、枝番Bは実表C153の同一「製品番号+枝番A」内で最大の枝番Bに、それぞれ対応している。仮想表A171は、アプリケーション22から見れば、常にRDB15の最新状態を反映したインデックスの役目をしており、アプリケーション22から参照される。

【0063】アプリケーション22は、RDB15の最新データを取得するために、ストアプロシージャ17が定義生成する仮想表A171と関係づけを行った仮想表B221を定義してアクセス要求をRDBMS16に発行する。発行した要求に対応する処理結果がRDBMS16から返され、取得したデータを基にデータ処理を行う。

【0064】仮想表B221は、仮想表A171と実表A151と実表B152と実表C153を基にアプリケーション22が定義生成する仮想表であり、アプリケーション22が必要とするデータから成る。仮想表B221の構成を図7(d)に示す。仮想表B221は製品番号と製品名称と受付番号と障害内容とから成り、製品番号は仮想表A171の製品番号に対応する実表A151の製品番号に、製品名称は仮想表A171の製品番号に対応する実表A151の製品名称に、受付番号は仮想表A171の「製品番号+枝番A」に対応する実表B152の受付番号に、障害内容は仮想表A171の「製品番号+枝番A+枝番B」に対応する実表C153の障害内容に、それぞれ対応している。

【0065】本発明の第2の実施の形態の動作について、図6～図9を参照して詳細に説明する。

【0066】図8は第2の実施の形態の動作を説明する図で、(a)は実表A151の具体例を、(b)は実表B152の具体例を、(c)は実表C153の具体例を、(d)は仮想表A171の具体例を、(e)は仮想表B221の具体例を、示したものである。(b)において、受付番号“002”および“003”は、一旦受け付けられたがその後削除されたので、削除に“DEL”と示されている。図9は第2の実施の形態の動作の流れを示す図である。

【0067】図9を参照すると、アプリケーション22は、実表A151の製品番号と仮想表A171の製品番号とを関係づけ、実表B152の「製品番号+枝番A」と仮想表A171の「製品番号+枝番A」とを関係づけ、実表C153の「製品番号+枝番A+枝番B」と仮想表A171の「製品番号+枝番A+枝番B」とを関係づけ、製品番号と製品名称と受付番号と障害内容とから成る仮想表B221を定義して、RDB15に対するアクセスをRDBMS16に要求する(ステップA21～A22)。

【0068】RDBMS16は、アプリケーション22からのRDB15に対するアクセス要求を分析し(ステップM21)、ストアプロシージャ17が定義生成する仮想表A171が指定されているときは、ストアプロシージャ17の起動実行が必要であるか否かを判断し、ストアプロシージャ17の起動実行が必要であることを認識したときにストアプロシージャ17を起動する(ステップM22～M24)。指定されたストアプロシージャ17が既に起動実行されていて且つその後

にRDB15の更新がなされていない場合には、ストアプロシージャ17を起動しない。この場合、既に実行されたストアプロシージャ17の処理結果が使用される。

【0069】起動されたストアプロシージャ17は以下の処理を実行する。

(1) 実表C153の「製品番号+枝番A」ごとに、同一「製品番号+枝番A」内で枝番Bが最大のものを含む行を選択する(ステップS21)。この例では、図8(c)の右端に○印が付いている行が選択される。

(2) 実表B152の「製品番号+枝番A」と上記(1)の結果の「製品番号+枝番A」とを関係づける(ステップS22)。このとき、実表B152の削除に“DEL”が設定されている実表B152の「製品番号+枝番A」は除く。この例では、受付番号“002”および“003”の行は削除に“DEL”が設定されているので除かれ、関係づけられるのは、図8(b)および(c)の右端に◎印が付いている行である。

(3) 実表A151の製品番号と上記(2)の結果の枝番Aと上記(2)の結果の枝番Bとから成る仮想表A171のデータセットを作成する(ステップS23)。ストアプロシージャ17の出力結果を図8(d)に示す。

【0070】続いて、RDBMS16は仮想表B221に基づく処理を行う(ステップM25)。RDBMS16の処理結果を図8(e)に示す。

【0071】その後、RDBMS16は処理結果をアプリケーション22に返す(ステップM26)。

【0072】アプリケーション22は、RDBMS16から処理結果を受け取り、アプリケーション処理を行う(ステップA23)。

【0073】このようにして、アプリケーション22は、ストアプロシージャ17の出力結果である仮想表A171をインデックス代わりに使用することにより、求めているRDB15の最新のデータを高速に得ることができる。すなわち、アプリケーション22は複雑な処理をすることなく、製品番号に対する最新の受付番号と障害内容とを取得して、処理を行うことができる。

【0074】本発明による上述した実施の形態において、データベースアクセス方式の処理動作を実行するためのプログラム等を、データとして磁気ディスクや光ディスク等の記憶装置(図示せず)に記憶するようにし、記憶されたデータを読み出してデータベースアクセス方式を動作させるために用いる。このように、本発明によるデータベースアクセス方式を動作させるデータを記憶媒体に記憶させ、この記憶媒体をインストールすることによりデータベースアクセス方式の機能が実現できるようになる。

【0075】

【発明の効果】第1の効果は、必要なストアプロシ

13

ジャを参照するだけで、アプリケーションから常に最新のデータを高速に参照することができることである。その理由は、アプリケーション用のインデックスを生成する機能をストアードプロシージャの中に構成するような手段を設けたためである。

【0076】第2の効果は、クライアントの負荷が軽減され且つネットワークのトラフィックも減少することである。その理由は、ストアードプロシージャがサーバー側で実行されるので、アプリケーションから複雑なSQLを発行する必要がなく、アプリケーションも一意キーの表を基にしたデータ操作処理を記述するだけで処理が構築でき、クライアントのアプリケーションとサーバーのRDBMSとの間に発生するやりとりも少なく済むためである。

【0077】第3の効果は、データベースの障害時、アプリケーション用のインデックスのために物理的なインデックステーブルの再作成といった特別の復旧処置が要らないことである。その理由は、アプリケーション用のインデックスが仮想的なテーブルとしてストアードプロシージャを利用しているためである。

【図面の簡単な説明】

【図1】本発明を説明する図

【図2】第1の実施の形態の構成を示す図

【図3】第1の実施の形態における(a)実表A(b)実表B(c)仮想表A(d)仮想表Bの構成を示す図

【図4】第1の実施の形態の動作を説明する(a)実表A(b)実表B(c)仮想表A(d)仮想表Bの図

【図5】第1の実施の形態の動作の流れを示す図

【図6】第2の実施の形態の構成を示す図

【図7】第2の実施の形態における(a)実表A(b)実表B(c)実表C(d)仮想表A(e)仮想表Bの構成を示す図

14

【図8】第2の実施の形態の動作を説明する(a)実表A(b)実表B(c)実表C(d)仮想表A(e)仮想表Bの図

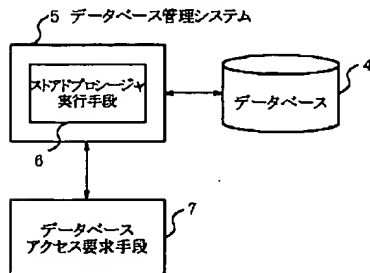
【図9】第2の実施の形態の動作の流れを示す図

【図10】従来の動作を説明する(a)実表A(b)実表B(c)結果の図

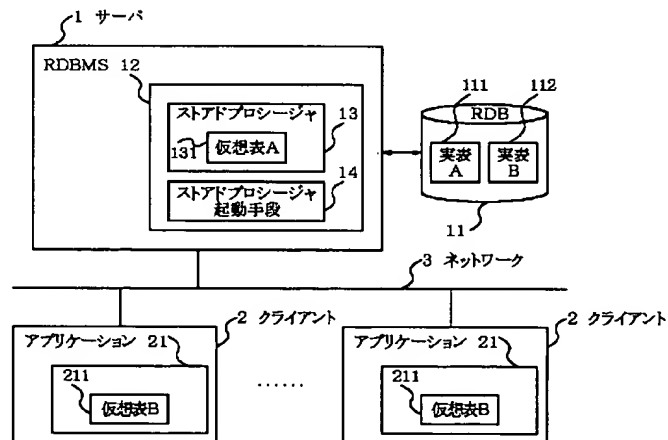
【符号の説明】

- | | |
|-----|-----------------|
| 1 | サーバ |
| 2 | クライアント |
| 3 | ネットワーク |
| 4 | データベース |
| 5 | データベース管理システム |
| 6 | ストアードプロシージャ実行手段 |
| 7 | データベースアクセス要求手段 |
| 11 | RDB |
| 12 | RDBMS |
| 13 | ストアードプロシージャ |
| 14 | ストアードプロシージャ起動手段 |
| 15 | RDB |
| 16 | RDBMS |
| 17 | ストアードプロシージャ |
| 21 | アプリケーション |
| 22 | アプリケーション |
| 111 | 実表A |
| 112 | 実表B |
| 131 | 仮想表A |
| 151 | 実表A |
| 152 | 実表B |
| 153 | 実表C |
| 171 | 仮想表A |
| 211 | 仮想表B |
| 221 | 仮想表B |

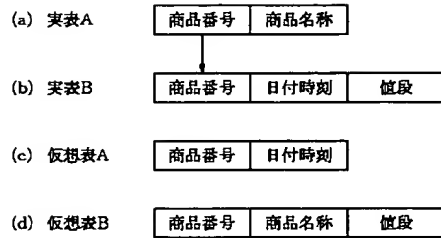
【図1】



【図2】



【図3】



【図4】

(a) 実表A

商品番号	商品名称
1	あああ
2	いゐい
3	ううう
4	えええ
5	おおお

(b) 実表B

商品番号	日付時刻	値段
1	05261000	200
1	05261200	198
1	05261400	195
1	05261600	193
1	05261800	190
2	05261000	100
2	05261600	95
3	05261000	498
4	05261000	300
4	05261200	295
4	05261400	290
5	05261000	98
5	05261800	70

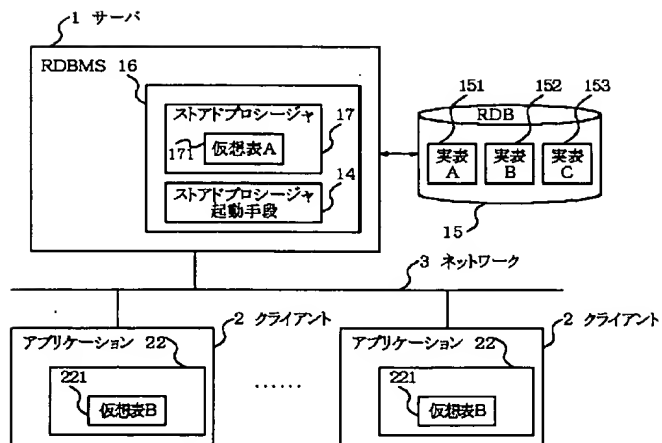
(c) 仮想表A

商品番号	日付時刻
1	05261800
2	05261800
3	05261000
4	05261400
5	05261800

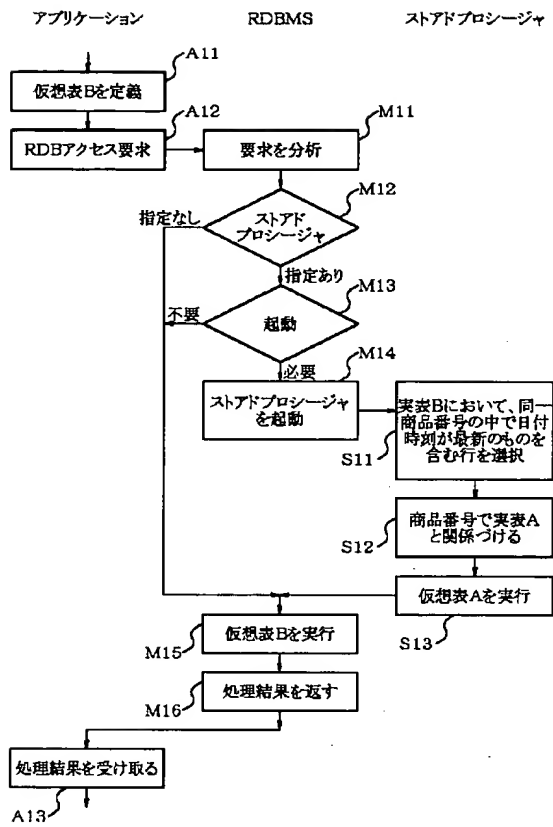
(d) 仮想表B

商品番号	商品名称	値段
1	あああ	190
2	いゐい	95
3	ううう	498
4	えええ	290
5	おおお	70

【図6】



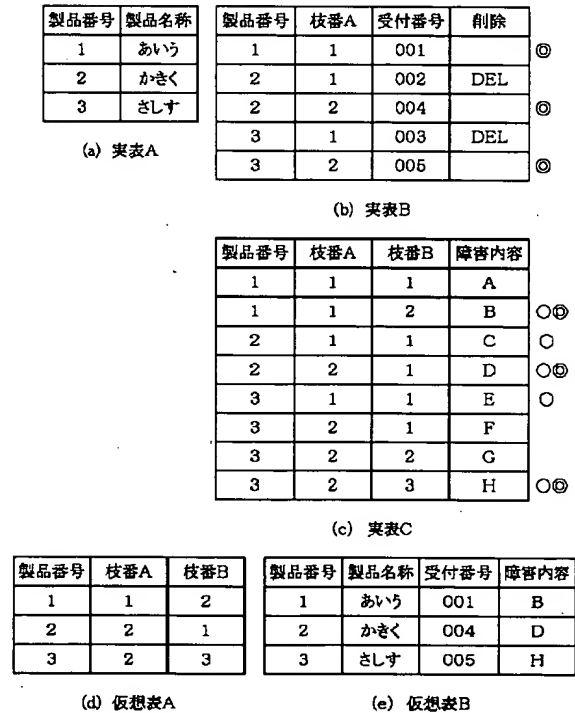
【図5】



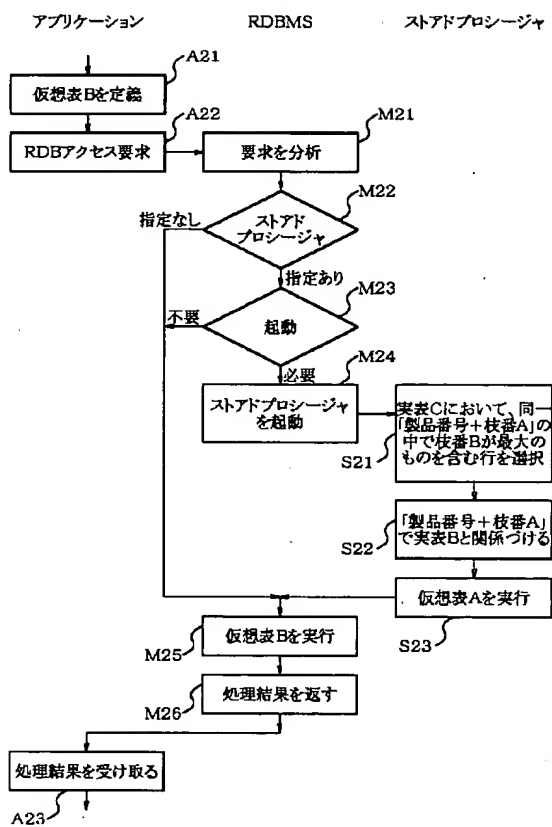
【図7】



【図8】



【図9】



【図10】

商品番号	商品名称
1	あああ
2	いはい
3	ううう
4	えええ
5	おおお

(a) 実表A

商品番号	日付時刻	値段
1	05261000	200
1	05261200	198
1	05261400	195
1	05261600	193
1	05261800	190
2	05261000	100
2	05261600	95
3	05261000	498
4	05261000	300
4	05261200	295
4	05261400	290
5	05261000	98
5	05261800	70

(b) 実表B

商品番号	商品名称	値段
1	あああ	190
2	いはい	95
3	ううう	498
4	えええ	290
5	おおお	70

(c) 結果

JP 2001 005 704 A

Pub. 12 Jan 2001

APPN JP 11172028

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] This invention relates to the data base access method, method, and storage which access a data base using the stored procedure registered especially into the data base management system about the data base access method, method, and storage which access a data base using a data base management system.

[0002]

[Description of the Prior Art] With the relational database management system (it is hereafter called RDBMS for short) which is one of the data base management systems, the relational database (it is hereafter called RDB for short) is managed by the tabular format. The real table and data with which data is actually stored have in a table the virtual table (it is also called a view) generated from the data of the real table which is not saved but becomes origin at the time of activation. Application acquires the required data on RDB through RDBMS using the SQL language which refers to a table.

[0003] In the client/server system (it is hereafter called a C/S system for short), RDB is connected to a server and it has RDBMS, and the application which exists in a client side performs a RDB access request to RDBMS using SQL language, and has become the gestalt which gets an access result from RDBMS.

[0004] That is, when searching the data which needs the application of a client, application publishes retrieval conditional expression united with the format of SQL language to RDBMS of a server, chooses the data corresponding to the retrieval conditions to which RDBMS searched the data of RDB and was given, and returns a retrieval result to the application of a client. At this time, rapid access techniques, such as creation of the index for retrieval, are used in RDBMS.

[0005] However, the index created by RDBMS is not necessarily created in the optimal form for application.

[0006] For example, in RDB which consists of the real table A as shown in drawing 10 (a) and (b), and the real table B, the application which asks for the newest price for every goods is considered.

[0007] Application performs the following processings.

- (1) Choose the line in which the date time of day contains the newest thing in the same quotient lot number number in the real table B.
- (2) Connect the result of the real table A and the above (1) in each quotient lot number number.
- (3) Relating of the above (2) generates a quotient lot number number, a goods name, and the table that consists of a price.
- (4) Access the table which it is as a result of the above (3), and acquire the data of a quotient lot number number, a goods name, and a price.

[0008] In this case, although the index of the date time of day is created in RDBMS, the index to the thing of the newest [time of day / date] is not created in the same quotient lot number number. For this reason, application must describe processing procedure for the date time of day to choose the

newest thing in the above (1) in the same quotient lot number number. Thus, the index which RDBMS creates is not the the best for each application.

[0009] Moreover, since the above-mentioned application process is performed by each client, information will go back and forth frequently between a server and a client through a network the result from the demand information and the server from each client.

[0010]

[Problem(s) to be Solved by the Invention] In the Prior art mentioned above, the index created by RDBMS is a simple index of a data item, and has the trouble that the data needed especially by the application side does not become index format, but creation of application becomes complicated. moreover -- therefore, there is also a trouble that the traffic between RDBMS and application increases.

[0011] Using the stored procedure registered into RDBMS, the purpose of this invention creates a required index for every application, simplifies application creation, and is to offer the means which raises the system-wide engine performance.

[0012]

[Means for Solving the Problem] A data base access method of invention of this application 1st A data base which carries out storage are recording of the data, and a database management system which receives a demand from a data base access request means, accesses said data base, and returns an access result to said data base access request means, A stored procedure activation means to create a table of a meaning key which was contained in said database management system, performed a stored procedure, and was defined as a stored procedure, It has a data base access request means to give a demand which accesses said data base with reference to a table of a meaning key which said stored procedure activation means created to said database management system.

[0013] A data base access method of invention of this application 2nd In a client/server system by which a client is connected with a server through a network said server A relational database which carries out storage are recording of the data, and consists of one or more real tables, and a rational database management system which accesses said relational database based on processing demanded from application, and returns a processing result to said application, A stored procedure which defines a virtual table connected with said real table, and creates a table of a meaning key, It has a stored procedure starting means to start said stored procedure. Said client It has application which publishes a demand which specifies a virtual table which said stored procedure defines, and accesses said relational database to said rational database management system, and receives a demand result from said rational database management system.

[0014] A data base access method of invention of this application 3rd is characterized by not starting said stored procedure, when, as for said stored procedure starting means, renewal of said relational database is not made after that by already carrying out starting activation of said stored procedure in the 2nd invention.

[0015] A virtual table where said stored procedure defines a data base access method of invention of this application 4th in the 2nd invention is characterized by carrying out a duty of an index, when application accesses said relational database.

[0016] A data base access method of invention of this application 5th It is the method of accessing a data base in a client/server system by which a client is connected with a server through a network. Application defines a table which connected a virtual table defined by a real table and stored procedure, and performs an access request to a relational database to a rational database management system. Said rational database management system analyzes an access request from said application. A virtual table which carried out starting activation of the stored procedure, and was defined as a stored procedure when a virtual table said stored procedure carries out [a virtual table] definition generation was specified is generated. An access request of application which used said virtual table is processed, a processing result is returned to application, and application is characterized by receiving a processing result from a rational database management system.

[0017] A data base access method of invention of this application 6th is characterized by said stored procedure being processing which generates a virtual table which consists of a meaning key of hysteresis of a real table of the master data, and two real tables or more holding the hysteresis to the newest in the 5th invention.

[0018] Access request processing which a record medium of invention of this application 7th defines a table with which application connected a virtual table defined by a real table and stored procedure, and performs an access request to a relational database to a rational database management system, Said rational database management system analyzes an access request from said application. Stored procedure starting processing in which said stored procedure is started when a virtual table said stored procedure carries out [a virtual table] definition generation is specified, Virtual table generation processing which generates a virtual table which performed said stored procedure and was defined as a stored procedure, Access processing which processes an access request of application which used said virtual table, and returns a processing result to application, It is characterized by recording a program for making a computer perform data receipt processing in which application receives a processing result from said rational database management system.

[0019]

[Embodiment of the Invention] The data base access method of this invention is explained with reference to drawing 1 .

[0020] Reference of drawing 1 constitutes the data base access method of this invention from the data base 4, a database management system 5, a stored procedure activation means 6, and a data base access request means 7.

[0021] A data base 4 is a data base which carries out storage are recording of the data.

[0022] A database management system 5 manages a data base 4, accesses a data base 4 based on the demand from the data base access request means 7, and returns an access result to the data base access request means 7.

[0023] The stored procedure activation means 6 performs the stored procedure registered into the database management system 5. The stored procedure includes the processing which creates the table (index) of the meaning key which the data base access request means 7 uses.

[0024] The data base access request means 7 requires a database management system 5 to perform the stored procedure activation means 6 and to access a data base 4 using an activation result, and receives a demand result from a database management system 5.

[0025] Actuation of this invention is explained. In addition, beforehand, the stored procedure which the data base access request means 7 uses shall be a well-known method, and shall be registered into the database management system 5.

[0026] First, the data base access request means 7 specifies the table (index) of the meaning key which a stored procedure creates, and sends the access request to a data base 4 to a database management system 5.

[0027] A database management system 5 performs access request processing specified after creating the table (index) of the meaning key which performs the stored procedure specified using the stored procedure activation means 6, and the data base access request means 7 uses, and returns a processing result to the data base access request means 7.

[0028] The data base access request means 7 receives a processing result from a database management system 5.

[0029] Thus, the data base access request means 7 can acquire the data for which it asks from a data base 4 at a high speed by creating the table (index) of the meaning key which the data base access request means 7 uses in the stored procedure registered into the database management system 5.

[0030] The gestalt of operation of this invention which applied the data base access method of this invention is explained.

[0031] The gestalt of operation of the 1st of this invention is explained to details with reference to a drawing. The gestalt of the 1st operation searches RDB which has managed the newest price of the

goods which consist of two real tables, and is related with the system which acquires and processes the newest price of goods.

[0032] Drawing 2 is drawing showing the 1st configuration of the gestalt of operation.

[0033] As for the gestalt of the 1st operation, reference of drawing 2 connects the client 2 with the server 1 in the network 3. The server 1 connected RDB11 and is equipped with RDBMS12. RDBMS12 includes a stored procedure 13 and the stored procedure starting means 14. A client 2 contains application 21.

[0034] RDB11 is the relational database which carries out storage and recording of the data, and consists of the real table A111 and the real table B112. The real table A111 is the master data, and the real table B112 holds the hysteresis of the master data.

[0035] The real table A111 is one of the tables which constitute RDB11. The configuration of the real table A111 is shown in drawing 3 (a). The real table A111 consists of a quotient lot number number and a goods name, and each line is distinguished by the meaning by the quotient lot number number which is a major key.

[0036] The real table B112 is one of the tables which constitute RDB11. The configuration of the real table B112 is shown in drawing 3 (b). The real table B112 consists of a quotient lot number number, the date time of day, and a price, and shows the newest price of goods. The line which shows the date time of day changed whenever the price of goods was changed, and a new price is registered. The quotient lot number number corresponds with the quotient lot number number of the real table A111, and, thereby, the real table A111 and the real table B112 are connected. The date time of day is the date and time of day when a price is registered, and shows the registration sequence of the price within the same quotient lot number number. It may change at the date time of day, and the sequence number may be used. Each line is uniquely distinguished by what connected the date time of day with the quotient lot number number (it is hereafter described as "quotient lot number number + date time of day").

[0037] RDBMS12 manages and accesses RDB11. The demand published from application 21 is processed and the processing result corresponding to the published demand is returned to application 21.

[0038] A stored procedure 13 performs processing which definition-generation-carries out selection creation of the data set for a virtual table A131 based on the real table A111 and the real table B112. A stored procedure 13 is started by the stored procedure starting means 14. The identifier of a virtual table A131 is determined when registering a stored procedure 13 into RDBMS12. Refer to the virtual table A131 which the stored procedure 13 generated for application 21 by specifying this identifier. In addition, about the method of registering a stored procedure 13 into RDBMS12, it is based on a well-known method.

[0039] A virtual table A131 is a virtual table which consists of the meaning key of the newest hysteresis. The selection condition with which a stored procedure 13 combines the newest (max) date time of day in the quotient lot number number of the real table B112 corresponding to [definition generation] the quotient lot number number of the real table A111 and it based on the real table A111 and the real table B112 is specified. The configuration of a virtual table A131 is shown in drawing 3 (c). A virtual table A131 consists of a quotient lot number number and the date time of day, and that of the real table A111 supports [the quotient lot number number] the quotient lot number number, respectively at the date time of day of the newest [time of day / date] in the same quotient lot number number of the real table B112 (max). If it sees from application 21, the virtual table A131 will carry out the duty of the index which always reflected the newest condition of RDB11, and will be referred to from application 21.

[0040] The stored procedure starting means 14 starts a stored procedure 13, when the demand whose application 21 uses the virtual table A131 which a stored procedure 13 generates is given to RDBMS12. When starting activation of the specified stored procedure 13 has already been carried out and renewal of RDB11 is not made after that, a stored procedure 13 is not started. In this case, the

processing result of the already performed stored procedure 13 is used.

[0041] In order to acquire the newest data of RDB11, application 21 defines the virtual table A131 as for which a stored procedure 13 carries out definition generation, and the connected virtual table B211, and publishes an access request to RDBMS12. The processing result corresponding to the published demand is returned from RDBMS12, and data processing is performed based on the acquired data.

[0042] Application 21 is a definition generation virtual table based on a virtual table A131, the real table A111, and the real table B112, and a virtual table B211 consists of the data which application 21 needs. The configuration of a virtual table B211 is shown in drawing 3 (d). A virtual table B211 consists of a quotient lot number number, a goods name, and a price, and a quotient lot number number in the quotient lot number number of the real table A111 corresponding to the quotient lot number number of a virtual table A131. The goods name supports the price of the real table B112 corresponding to the "quotient lot number number + date time of day" of a virtual table A131 in a price at the goods name of the real table A111 corresponding to the quotient lot number number of a virtual table A131, respectively.

[0043] Actuation of the gestalt of operation of the 1st of this invention is explained to details with reference to drawing 1 - drawing 5.

[0044] drawing 4 explains actuation of the gestalt of the 1st operation -- it is -- (a) -- the example of the real table A111 -- in (b), (c) shows the example of a virtual table A131, and (d) shows the example of a virtual table B211 for the example of the real table B112. the goods name from (a) and (b) -- " -- such -- the newest price of *" -- the date time of day -- it turns out that it is "190" set as "05261800." Drawing 5 is drawing showing the flow of actuation of the gestalt of the 1st operation.

[0045] When drawing 5 is referred to, application 21 The quotient lot number number of the real table A111 and the quotient lot number number of a virtual table A131 are connected. The "quotient lot number number + date time of day" of the real table B112 and the "quotient lot number number + date time of day" of a virtual table A131 are connected, the virtual table B211 which consists of a quotient lot number number, a goods name, and a price is defined, and access to RDB11 is required of RDBMS12 (steps A11-A12).

[0046] It judges whether RDBMS12 needs starting activation of a stored procedure 13, when the virtual table A131 analyzes the access request to RDB11 from application 21 (step M11), and a stored procedure 13 carries out [the virtual table] definition generation is specified, and when it has recognized that a stored procedure 13 needs to be starting performed, a stored procedure 13 is started (steps M12-M14). When starting activation of the specified stored procedure 13 has already been carried out and renewal of RDB11 is not made after that, a stored procedure 13 is not started. In this case, the processing result of the already performed stored procedure 13 is used.

[0047] The started stored procedure 13 performs the following processings.

(1) The date time of day chooses the line containing the newest thing within the same quotient lot number number for every quotient lot number number of the real table B112 (step S11).

(2) Connect the quotient lot number number of the real table A111, and the quotient lot number number as a result of the above (1) (step S12).

(3) Create the quotient lot number number of the real table A111, and the data set of the virtual table A131 which consists of the date time of day as a result of the above (1) (step S13). The output of a stored procedure 13 is shown in drawing 4 (c).

[0048] Then, RDBMS12 performs processing based on a virtual table B211 (step M15). The processing result of RDBMS12 is shown in drawing 4 (d).

[0049] Then, RDBMS12 returns a processing result to application 21 (step M16).

[0050] Application 21 performs reception and an application process for a processing result from RDBMS12 (step A13).

[0051] Thus, application 21 can obtain the newest data of RDB11 for which it is asking at a high speed by using the virtual table A131 which is the output of a stored procedure 13 instead of an index.

[0052] next, gestalt of operation of the 2nd of this invention ***** -- it explains to details with

reference to a drawing. The gestalt of the 2nd operation searches RDB which has managed the contents of a failure of the product which consists of three real tables, and is related with the system which acquires and processes the newest contents of a failure of a product.

[0053] Drawing 6 is drawing showing the 2nd configuration of the gestalt of operation.

[0054] If drawing 6 is referred to, as for the gestalt of the 2nd operation, a client 2 will be connected with a server 1 in a network 3. A server 1 is equipped with RDBMS16 which connects RDB15 and includes a stored procedure 17 and the stored procedure starting means 14. A client 2 is equipped with application 22 and constituted. RDB11, RDBMS12, the stored procedure 13, and application 21 of a gestalt of the 1st operation It is replaced by RDB15, RDBMS16, the stored procedure 17, and application 22.

[0055] Here, RDB15, RDBMS16, the different stored procedure 17, and different application 22 from the gestalt of the 1st operation are explained.

[0056] RDB15 is the relational database which carries out storage are recording of the data, and consists of the real table A151, the real table B152, and the real table C153. The real table A151 is the master data, and the real table B152 and the real table C153 hold the hysteresis of the master data.

[0057] The real table A151 is one of the tables which constitute RDB15. The configuration of the real table A151 is shown in drawing 7 (a). The real table A151 consists of a part number and a product name, and each line is distinguished by the meaning by the part number which is a major key.

[0058] The real table B152 is one of the tables which constitute RDB15. The configuration of the real table B152 is shown in drawing 7 (b). The real table B152 consists of a part number, a branch number A and a receipt number, and deletion. The part number corresponds with the part number of the real table A151, and, thereby, the real table B152 and the real table A151 are connected. A branch number A may show the sequence of the receipt number within the same part number, and the sequence number or the date time of day is sufficient as it. Each line is uniquely distinguished with "the part number + branch number A." A receipt number is the sequence number ****(ed) through a product, and a receipt number is registered whenever the failure of a product is received. The contents of the failure are registered into the real table C153. When there is new failure reception to the same product, a branch number A is raised and a new receipt number is registered. Deletion shows whether it was deleted once the receipt number was received, and when a receipt number is deleted, it is indicated to be "DEL."

[0059] The real table C153 is one of the tables which constitute RDB15. The configuration of the real table C153 is shown in drawing 7 (c). The real table C153 consists of a part number, and a branch number A, a branch number B and the contents of a failure. "The part number + branch number A" corresponds with the "part number + branch number A" of the real table B152, and, thereby, the real table C153 and the real table B152 are connected. A branch number B may show the sequence of the contents of a failure within the same "part number + branch number A", and the sequence number or the date time of day is sufficient as it. Each line is uniquely distinguished with "the part number + branch number A+ branch number B." The contents of a failure corresponding to a receipt number are registered into the contents of a failure. When the contents of a failure about the received receipt number are updated, the contents of a failure which raised the branch number B and were updated are registered, and hysteresis management is performed as another data.

[0060] RDBMS16 manages and accesses RDB15. The demand published from application 22 is processed and the processing result corresponding to the published demand is returned to application 22.

[0061] A stored procedure 17 performs processing which definition-generation-carries out selection creation of the data set for a virtual table A171 based on the real table A151, the real table B152, and the real table C153. A stored procedure 17 is started by the stored procedure starting means 14. The identifier of a virtual table A171 is determined when registering a stored procedure 17 into RDBMS16. Refer to the virtual table A171 which the stored procedure 17 generated for application 22 by specifying this identifier. In addition, about the method of registering a stored procedure 17 into

RDBMS16, it is based on a well-known method.

[0062] A virtual table A171 is a virtual table which consists of the meaning key of the newest hysteresis. The selection condition with which a stored procedure 17 definition-generation-combines the greatest branch number B within the greatest branch number A in the part number of the real table A151 and the same part number of the real table B152 and the same "part number + branch number A" of the real table C153 based on the real table A151, the real table B152, and the real table C153 is specified. The configuration of a virtual table A171 is shown in drawing 7 (c). A virtual table A171 consists of a part number, a branch number A, and a branch number B, a branch number A is equivalent to the greatest branch number A in the same part number of the real table B152, and the branch number B is equivalent to the greatest branch number B at the part number of the real table A151 for the part number, respectively within the same "part number + branch number A" of the real table C153. If it sees from application 22, the virtual table A171 will carry out the duty of the index which always reflected the newest condition of RDB15, and will be referred to from application 22.

[0063] In order to acquire the newest data of RDB15, application 22 defines the virtual table A171 as for which a stored procedure 17 carries out definition generation, and the connected virtual table B221, and publishes an access request to RDBMS16. The processing result corresponding to the published demand is returned from RDBMS16, and data processing is performed based on the acquired data.

[0064] Application 22 is a definition generation virtual table based on a virtual table A171, the real table A151, the real table B152, and the real table C153, and a virtual table B221 consists of the data which application 22 needs. The configuration of a virtual table B221 is shown in drawing 7 (d). A virtual table B221 consists of a part number, a product name, a receipt number, and the contents of a failure. To the part number of the real table A151 corresponding to the part number of a virtual table A171, a part number To the product name of the real table A151 corresponding to the part number of a virtual table A171, a product name The receipt number supports the contents of a failure of the real table C153 corresponding to the "part number + branch number A+ branch number B" of a virtual table A171 in the contents of a failure at the receipt number of the real table B152 corresponding to the "part number + branch number A" of a virtual table A171, respectively.

[0065] Actuation of the gestalt of operation of the 2nd of this invention is explained to details with reference to drawing 6 - drawing 9.

[0066] drawing where drawing 8 explains actuation of the gestalt of the 2nd operation -- it is -- (a) -- the example of the real table A151 -- (b) -- the example of the real table B152 -- in (c), (d) shows the example of a virtual table A171, and (e) shows the example of a virtual table B221 for the example of the real table C153. In (b), although receipt number "002" and "003" were once received, since they were deleted after that, they are indicated to be "DEL" to deletion. Drawing 9 is drawing showing the flow of actuation of the gestalt of the 2nd operation.

[0067] When drawing 9 is referred to, application 22 The part number of the real table A151 and the part number of a virtual table A171 are connected. The "part number + branch number A" of the real table B152 and the "part number + branch number A" of a virtual table A171 are connected. Connect the "part number + branch number A+ branch number B" of the real table C153, and the "part number + branch number A+ branch number B" of a virtual table A171, and the virtual table B221 which consists of a part number, a product name, a receipt number, and the contents of a failure is defined. Access to RDB15 is required of RDBMS16 (steps A21-A22).

[0068] It judges whether RDBMS16 needs starting activation of a stored procedure 17, when the virtual table A171 analyzes the access request to RDB15 from application 22 (step M21), and a stored procedure 17 carries out [the virtual table] definition generation is specified, and when it has recognized that a stored procedure 17 needs to be starting performed, a stored procedure 17 is started (steps M22-M24). When starting activation of the specified stored procedure 17 has already been carried out and renewal of RDB15 is not made after that, a stored procedure 17 is not started. In this case, the processing result of the already performed stored procedure 17 is used.

[0069] The started stored procedure 17 performs the following processings.

(1) the real table C153 -- "-- a branch number B chooses the line containing the greatest thing as every part number + branch number A" within the same "part number + branch number A" (step S21). The line to which O mark is attached is chosen as the right end of drawing 8 (c) in this example.

(2) Connect the "part number + branch number A" of the real table B152, and the "part number + branch number A" as a result of the above (1) (step S22). At this time, the "part number + branch number A" of the real table B152 with which "DEL" is set as deletion of the real table B152 is removed. In this example, that remove and receipt number "002" and the line of "003" are connected since "DEL" is set as deletion is a line to which drawing 8 (b) and the right end of (c) have O mark.

(3) Create the data set of the virtual table A171 which consists of the part number of the real table A151, the branch number A as a result of the above (2), and the branch number B as a result of the above (2) (step S23). The output of a stored procedure 17 is shown in drawing 8 (d).

[0070] Then, RDBMS16 performs processing based on a virtual table B221 (step M25). The processing result of RDBMS16 is shown in drawing 8 (e).

[0071] Then, RDBMS16 returns a processing result to application 22 (step M26).

[0072] Application 22 performs reception and an application process for a processing result from RDBMS16 (step A23).

[0073] Thus, application 22 can obtain the newest data of RDB15 for which it is asking at a high speed by using the virtual table A171 which is the output of a stored procedure 17 instead of an index.

Namely, application 22 can be processed by acquiring the newest receipt number and the newest contents of a failure over a part number, without carrying out complicated processing.

[0074] In the gestalt of the operation by this invention mentioned above, it uses in order to memorize the program for performing processing actuation of a data base access method etc. to storage (not shown), such as a magnetic disk and an optical disk, as data, to read the memorized data and to operate a data base access method. Thus, the data which operates the data base access method by this invention is stored in a storage, and the function of a data base access method can be realized now by installing this storage.

[0075]

[Effect of the Invention] The 1st effect is only referring to a required stored procedure and always being able to refer to the newest data at a high speed from application. The reason is because a means which constitutes the function which generates the index for applications in a stored procedure was established.

[0076] The 2nd effect is that the load of a client is mitigated and network traffic also decreases. Since a stored procedure is performed by the server side, the reason is for there being also few exchanges which do not need to publish complicated SQL from application, can build processing only by application describing the data manipulation processing based on the table of a meaning key, and are generated between the application of a client and RDBMS of a server, and ending.

[0077] The 3rd effect is that special restoration treatment called re-creation of an index table physical for the index for applications is not needed at the time of the failure of a data base. The reason is because the stored procedure is used as a table with the imagination index for applications.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] A data base access method characterized by providing the following A data base which carries out storage are recording of the data A database management system which receives a demand from a data base access request means, accesses said data base, and returns an access result to said data base access request means A stored procedure activation means to create a table of a meaning key which was contained in said database management system, performed a stored procedure, and was defined as a stored procedure A data base access request means to give a demand which accesses said data base with reference to a table of a meaning key which said stored procedure activation means created to said database management system

[Claim 2] A client/server system which is characterized by providing the following and by which a client is connected with a server through a network Said server is a relational database which carries out storage are recording of the data, and consists of one or more real tables. A rational database management system which accesses said relational database based on processing demanded from application, and returns a processing result to said application A stored procedure which defines a virtual table connected with said real table, and creates a table of a meaning key It is the application which is equipped with a stored procedure starting means to start said stored procedure, publishes a demand which said client specifies a virtual table which said stored procedure defines, and accesses said relational database to said rational database management system, and receives a demand result from said rational database management system.

[Claim 3] Said stored procedure starting means is a data base access method according to claim 2 which starting activation of said stored procedure is already carried out, and is characterized by not starting said stored procedure when renewal of said relational database is not made after that.

[Claim 4] A virtual table which said stored procedure defines is a data base access method according to claim 2 characterized by carrying out a duty of an index when application accesses said relational database.

[Claim 5] It is the method of accessing a data base in a client/server system by which a client is connected with a server through a network. Application defines a table which connected a virtual table defined by a real table and stored procedure, and performs an access request to a relational database to a rational database management system. Said rational database management system analyzes an access request from said application. A virtual table which carried out starting activation of the stored procedure, and was defined as a stored procedure when a virtual table said stored procedure carries out [a virtual table] definition generation was specified is generated. It is the data base access method which processes an access request of application which used said virtual table, returns a processing result to application, and is characterized by application receiving a processing result from a rational database management system.

[Claim 6] Said stored procedure is the data base access method according to claim 5 characterized by being the processing which generates a virtual table which consists of a meaning key of the newest

hysteresis from a real table of the master data, and two real tables or more holding the hysteresis.
[Claim 7] Access request processing which defines a table with which application connected a virtual table defined by a real table and stored procedure, and performs an access request to a relational database to a rational database management system, Said rational database management system analyzes an access request from said application. Stored procedure starting processing in which said stored procedure is started when a virtual table said stored procedure carries out [a virtual table] definition generation is specified, Virtual table generation processing which generates a virtual table which performed said stored procedure and was defined as a stored procedure, Access processing which processes an access request of application which used said virtual table, and returns a processing result to application, A record medium characterized by recording a program for making a computer perform data receipt processing in which application receives a processing result from said rational database management system.

[Translation done.]

PTO 04-1710

Japanese Kokai Patent Application
No. 2001-5704

DATABASE ACCESS SYSTEM, METHOD, AND STORAGE MEDIUM

Sunao Koderu

seems to apply
2/17/04

UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. FEBRUARY 2004
TRANSLATED BY THE RALPH MCELROY TRANSLATION COMPANY

JAPANESE PATENT OFFICE
PATENT JOURNAL
KOKAI PATENT APPLICATION NO. 2001-5704

Int. Cl. ⁷: G 06 F 12/00
17/30

Identification Code: 513

Theme Codes (References): 5B075
5B082

Filing No.: Hei 11[1999]-172028

Filing Date: June 18, 1999

Publication Date: January 12, 2001

Examination Request: Requested

No. of Claims: 7 (Total of 11 pages)

DATABASE ACCESS SYSTEM, METHOD, AND STORAGE MEDIUM
[Deetabeesu akusesu hoshiki, hoho oyobi kioku baitai]

Inventor: Sunao Koderu

Applicant: 000211329
Chugoku NEC Software, Ltd.

F Terms (References): 5B075 NK02 NK10 NK24 NK54
PQ02 PQ05 5B082 BA09 EA05
GA08

[There are no amendments to this patent.]

* * *

Claims

1. A database access system characterized in that it is equipped with a database for storing data, a database management system which gains access to the aforementioned database upon receiving a request from a database access request means and returns the access results to the aforementioned database access request means, a stored procedure execution means contained in the aforementioned database management system which executes a stored procedure in order to generate a table of unique keys defined by the stored procedure, and a database access request means which issues a request to gain access to the aforementioned database to the aforementioned database management system in reference to the table of unique keys generated by the aforementioned stored procedure execution means.

2. A database access system characterized in that in a client server system in which a server and clients are connected via a network, the aforementioned server is equipped with a relational database which comprises more than 1 base tables containing data, a relational database management system which gains access to the aforementioned relational database according to a processing requested by an application program and returns the access results to the aforementioned application program, a stored procedure which defines a virtual table correlated with the aforementioned base tables in order to generate a table of unique keys, and a stored procedure activation means which activates the aforementioned stored procedure; and the aforementioned client is provided with an application program used to specify the virtual table defined by the aforementioned stored procedure in order to issue a request to gain access to the aforementioned relational database and to the aforementioned relational database management system and receive the results of the request from the aforementioned relational database management system.

3. A database access method described under Claim 2 characterized in that the aforementioned stored procedure activation means does not activate the aforementioned stored procedure when the aforementioned stored procedure was already activated and executed, and the aforementioned relational database has not been renewed since then.

4. The database access method described under Claim 2 characterized in that the virtual table defined by the aforementioned stored procedure serves the role of indexes used when the application program gains access to the aforementioned relational database.

5. A database access method characterized in that it is a method for gaining access to a database in a client server system in which a server and clients are connected via a network; wherein, an application program defines a table, in which base tables and a virtual table defined by a stored procedure are correlated, and issues a request to gain access to a relational database to

* [Numbers in the margin indicate pagination in the foreign text.]

a relational database management system; the aforementioned relational database management system analyzes the access request from the aforementioned application program, activates/executes the stored procedure if the virtual table defined and generated in accordance with the aforementioned stored procedure is specified in order to generate the virtual table defined by the stored procedure, processes the access request from the application program which used the aforementioned virtual table, and returns the processing results to the application program; and the application program receives the processing results from the relational database management system.

6. The database access method described under Claim 5 characterized in that the aforementioned stored procedure refers to processing for generating a virtual table comprising unique keys for the latest history from 2 or more base tables comprising [at least] a base table for basic data and a base table for keeping its history.

7. A storage medium characterized in that it stores a program in order for a computer to execute access request processing during which an application program defines a table in which base tables and a virtual table defined by a stored procedure are correlated in order to issue a request to gain access to a relational database to a database management system, stored procedure activation processing during which the aforementioned database management system analyzes the access request from the aforementioned application program and activates the aforementioned stored procedure if the virtual table defined and generated in accordance with the aforementioned stored procedure is specified, virtual table generation processing during which the aforementioned stored procedure is executed in order to generate the virtual table defined by the stored procedure, access processing during which the access request from the application program which used the aforementioned virtual table is processed, and the processing results are returned to the application program, and data reception processing during which the application program receives the processing results from the aforementioned relational database management system.

Detailed explanation of the invention

0001

Technical field of the invention

The present invention pertains to a database access system in which access is gained to a database using a database management system and a method and a medium to this end. In particular, it pertains to a database access system in which access is gained to a database using a stored procedure registered to a database management system and a method and a medium to this end.

[0002]

Prior art

In the case of a relational database management system (will be abbreviated as RDBMS, hereinafter) as one type of database management system, the relational database (will be abbreviated as RDB, hereinafter) is managed in the form of tables. The tables include base tables which contain actual data and a virtual table (referred to as view also) which is generated from the data in an original base table at the time of the execution of an instruction but not used for storing data. An application program obtains necessary data in the RDB via the RDBMS using the SQL language used for making references to the tables.

[0003]

In a client/server system (will be abbreviated as C/S system, hereinafter), the server is equipped with an RDBMS and connected to an RDB, and an application program provided at the client's side issues an RDB access request to the RDBMS using the SQL language and receives the access results from the RDBMS.

[0004]

That is, in order for the application program of the client to retrieve necessary data, the application program issues a search condition formula in accordance with the SQL language format to the RDBMS of the server; and the RDBMS runs a data search in the RDB, selects data which match the search condition specified, and returns the search results to the application program of the client. At this time, a high-speed access technique, for example, generation of indexes for the search, is used by the RDBMS.

/3

[0005]

However, the indexes generated by the RDBMS are not always optimized for the application program.

[0006]

For example, an application program used for obtaining the latest prices of respective merchandises in an RDB comprising base table A and base table B shown in Figure 10 (a) and (b) will be examined.

[0007]

The application program carries out the following processing.

(1) It selects the lines showing the latest time stamps [literally; "date and time"] for those with the same merchandise numbers in base table B.

(2) It correlates base table A with the results of (1) above for the respective merchandise numbers.

(3) It generates a table comprising the merchandise numbers, merchandise names, and prices according to the correlation obtained in (2) above.

(4) It gains access to the table obtained as a result of (3) above in order to obtain data on the merchandise numbers, the merchandise names, and the prices.

[0008]

In this case, although indexes for the time stamps are generated by the RDBMS, no indexes for those with the latest time stamps among those with the same merchandises are generated. Thus, the program application must describe a processing procedure for selecting those with the latest time stamps among those with the same merchandise numbers in (1) above. As such, the indexes generated by the RDBMS are not optimized for an individual application program.

[0009]

In addition, because the aforementioned processing is executed by individual clients, request information from the respective clients and result information from the server are exchanged between the server and the clients in a complex manner via the network.

[0010]

Problems to be solved by the invention

In the aforementioned prior art, the indexes generated by the RDBMS are simple indexes based on data entries, and particular data needed by the application program are not generated in the index format, resulting in the application program becoming complicated. As a result, it creates another problem that the traffic between the RDBMS and the application program becomes busier.

[0011]

The purpose of the present invention is to present a means by which indexes needed by each application program are generated using a stored procedure registered to an RDBMS in

order to simplify the application program, so that the overall performance of the system can be improved.

[0012]

Means to solve the problems

The database access system as a first invention of the present patent application is equipped with a database for storing data, a database management system which gains access to the aforementioned database upon receiving a request from a database access request means and returns the access results to the aforementioned database access request means, a stored procedure execution means contained in the aforementioned database management system which executes a stored procedure in order to generate a table of unique keys defined by the stored procedure, and a database access request means which issues a request to gain access to the aforementioned database to the aforementioned database management system in reference to the table of unique keys generated by the aforementioned stored procedure execution means.

[0013]

In the case of the database access system as a second invention of the present patent application, in a client server system in which a server and clients are connected via a network, the aforementioned server is equipped with a relational database which comprises more than 1 base tables containing data, a relational database management system which gains access to the aforementioned relational database according to a processing requested by an application program and returns the access results to the aforementioned application program, a stored procedure which defines a virtual table correlated with the aforementioned base tables in order to generate a table of unique keys, and a stored procedure activation means which activates the aforementioned stored procedure; and the aforementioned client is provided with an application program used to specify the virtual table defined by the aforementioned stored procedure in order to issue a request to gain access to the aforementioned relational database to the aforementioned relational database management system and receive the results of the request from the aforementioned relational database management system.

[0014]

The database access system as a third invention of the present patent application is characterized in that in the second invention, the aforementioned stored procedure activation means does not activate the aforementioned stored procedure when the aforementioned stored procedure was already activated and executed, and the aforementioned relational database has not been renewed since then.

[0015]

The database access system as a fourth invention of the present patent application is characterized in that in the second invention, the virtual table defined by the aforementioned stored procedure serves the role of indexes used when the application program gains access to the aforementioned relational database.

[0016]

The database access method as a fifth invention of the present patent application is characterized in that it is a method for gaining access to a database in a client server system in which a server and clients are connected via a network; wherein, an application program defines a table, in which base tables and a virtual table defined by a stored procedure are correlated, and issues a request to gain access to a relational database to a relational database management system; the aforementioned relational database management system analyzes the access request from the aforementioned application program, activates/executes the stored procedure if the virtual table defined and generated in accordance with the aforementioned stored procedure is specified in order to generate the virtual table defined by the stored procedure, processes the access request from the application program which used the aforementioned virtual table, and returns the processing results to the application program; and the application program receives the processing results from the relational database management system.

[0017]

The database access method as a sixth invention of the present patent application is characterized in that in the fifth invention, the aforementioned stored procedure refers to processing for generating a virtual table comprising unique keys for the latest history from 2 or more base tables comprising [at least] a base table for basic data and a base table for keeping its history.

/4

[0018]

The storage medium as a seventh invention of the present patent application is characterized in that it stores a program in order for a computer to execute access request processing during which an application program defines a table in which base tables and a virtual table defined by a stored procedure are correlated in order to issue a request to gain access to a relational database to a database management system, stored procedure activation processing during which the aforementioned database management system analyzes the access request from the aforementioned application program and activates the aforementioned stored procedure if the virtual table defined and generated in accordance with the aforementioned stored procedure is

specified, virtual table generation processing during which the aforementioned stored procedure is executed in order to generate the virtual table defined by the stored procedure, access processing during which the access request from the application program which used the aforementioned virtual table is processed, and the processing results are returned to the application program, and data reception processing during which the application program receives the processing results from the aforementioned relational database management system.

[0019]

Embodiments of the invention

The database access system of the present invention will be explained in reference to Figure 1.

[0020]

In reference to Figure 1, the database access system of the present invention is configured with database 4, database management system 5, stored procedure execution means 6, and database access request means 7.

[0021]

Database 4 is a database for storing and accumulating data.

[0022]

Database management system 5 manages database 4, gains access to database 4 upon receiving a request from database access request means 7, and returns the access results to database access request means 7.

[0023]

Stored procedure execution means 6 executes a stored procedure registered to database management system 5. The stored procedure includes processing for generating a table of unique keys (indexes) used by database access request means 7.

[0024]

Database access request means 7 requests database management system 5 to activate stored procedure execution means 6 and gain access to database 4 using the execution result and receives the request results from database management system 5.

[0025]

Operations of the present invention will be explained. Here, assume that the stored procedure to be used by database access request means 7 has been registered to database management system 5 in advance using a well-known method.

[0026]

First, database access request means 7 specifies a table of unique keys (indexes) to be generated by the stored procedure and sends a request to gain access to database 4 to database management system 5.

[0027]

Database management system 5 lets stored procedure execution means 6 execute the specified stored procedure to generate the table of unique keys (indexes) to be used by database access request means 7, carries out a specified access request processing then, and returns the processing results to database access request means 7.

[0028]

Database access request means 7 receives the processing results from database management system 5.

[0029]

As described above, when the table of unique keys (indexes) used by database access request means 7 is generated by the stored procedure registered to database management system 5, database access request means 7 can obtain the data it needs from database 4 quickly.

[0030]

Embodiments of the present invention to which the database access system of the present invention is applied will be explained.

[0031]

A first embodiment of the present invention will be explained in detail in reference to figures. The first embodiment pertains to a system which runs a search in a RDB for managing the latest prices of merchandises using 2 base tables to obtain the latest prices of the merchandises for further processing.

[0032]

Figure 2 is a diagram illustrating the configuration of the first embodiment.

[0033]

In reference to Figure 2, in the first embodiment, server 1 and clients 2 are connected via network 3. Server 1 is equipped with RDBMS 12 and connected to RDB 11. RDBMS 12 contains stored procedure 13 and stored procedure activation means 14. Client 2 contains application program 21.

[0034]

RDB 11 is a relational database for storing/accumulating data, and comprises base table A 111 and base table B 112. Base table A 111 holds basic data, and base table B 112 keeps the history of the basic data.

[0035]

Base table A 111 is one of the tables which constitute RDB 11. Configuration of base table A 111 is shown in Figure 3 (a). Base table A 111 comprises merchandise numbers and merchandise names, and respective lines are distinguished from each other uniquely using the merchandise numbers as the primary keys.

[0036]

Base table B 112 is one of the tables which constitute RDB 11. Configuration of base table B 112 is shown in Figure 3 (b). Base table B 112 comprises merchandise numbers and time stamps and shows the latest prices of the merchandise. Every time a merchandise price is changed, a line showing a time stamp indicating the change of price and the new price is registered. The merchandise numbers correspond to the merchandise numbers in base table A 111, and base table A 111 and base table B 112 are correlated with each other as a result. The time stamps show the dates and the times those price are registered, and they show the order in which the prices of those with the same merchandise numbers are registered. Serial numbers may be used in place of the time stamps. The respective lines are distinguished from each other uniquely using the combination of a merchandise number and a time stamp (will be noted as "merchandise number + time stamp," hereinafter).

[0037]

RDBMS 12 manages RDB 11 and gains access to it. It processes a request issued from application program 21 and returns the processing results corresponding to the request issued to application program 21.

[0038]

Stored procedure 13 defines and generates virtual table A 131 based on base table A 111 and base table B 112 in order to select and generate a dataset. Stored procedure 13 is activated by stored procedure activation means 14. Name of virtual table A 131 is decided when stored procedure 13 is registered to RDBMS 12. Application program 21 can make reference to virtual table A 131 generated by stored procedure 13 by specifying said name. Furthermore, a well-known method is used as the method for registering stored procedure 13 to RDBMS 12.

[0039]

Virtual table A 131 is a virtual table comprising unique keys with the latest history. It is defined and generated by stored procedure 13 based on base table A 111 and base table B 112; wherein, selection conditions for combining the merchandise numbers in base table A 111 with the corresponding merchandise numbers in base table B 112 with the latest (greatest) time stamps are specified. Configuration of virtual table A 131 is shown in Figure 3 (c). Virtual table A 131 comprises merchandise numbers and time stamps; wherein, the merchandise numbers and the time stamps [sic] in base table A 111 correspond to the merchandise numbers and the latest (greatest) time stamps of those with the same merchandise numbers in base table B 112, respectively. From the viewpoint of application program 21, virtual table A 131 plays the role of indexes which constantly reflect the latest condition of RDB 11 when a reference is made from application program 21.

[0040]

Stored procedure activation means 14 activates stored procedure 13 when application program 21 sends a request to RDBMS 12 to use virtual table A 131 generated by stored procedure 13. If the specified stored procedure 13 was already activated and executed, and RDB 11 has not been renewed since then, stored procedure 13 is not activated. In this case, the processing results of stored procedure 13 already executed are used.

[0041]

Application program 21 defines virtual table B 211 correlated with virtual table A 131 defined and generated by stored procedure 13 and issues an access request to RDBMS 12 in

order to obtain the latest data from RDB 11. Processing results corresponding to the request issued are returned from RDBMS 12, and data processing is carried out based on the data obtained.

[0042]

Virtual table B 211 is a virtual table defined and generated by application program 21 based on virtual table A 131, base table A 111, and base table B 112, and it comprises the data needed by application program 21. Configuration of virtual table B 211 is shown in Figure 3 (d). Virtual table B 211 comprises merchandise numbers, merchandise names, and prices; wherein, the merchandise numbers, the merchandise names, and the prices correspond to the merchandise numbers in base table A 111 corresponding to the merchandise numbers in virtual table A 131, the merchandise names in base table A 111 corresponding to the merchandise numbers in virtual table A 131, the prices in base table B 112 corresponding to “merchandise number + time stamp” in virtual table A 131, respectively.

[0043]

Operations of the first embodiment of the present invention will be explained in detail in reference to Figure 1 through Figure 5.

[0044]

Figure 4 shows tables for illustrating the operations of the first embodiment; wherein, (a) shows a specific example of base table A 111, (b) shows a specific example of base table B 112, (c) shows a specific example of virtual table A 131, and (d) shows a specific example of virtual table B 211. From (a) and (b), it is clear that the latest price of the merchandise name “A A A” is “190” which is set at the time stamp “05261800.” Figure 5 is a flowchart illustrating the flow of the first embodiment.

[0045]

In reference to Figure 5, application program 21 correlates the merchandise numbers in base table A 111 with the merchandise numbers in virtual table A 131 as well as “merchandise number + time stamp” in base table B 112 with “merchandise number + time stamp” in virtual table A 131 to define virtual table B 211 comprising the merchandise numbers, the merchandise names, and the prices and requests access to RDB 11 to RDBMS 12 (Steps A11-A12).

[0046]

RDBMS 12 analyzes the RDB 11 access request from application program 21 (Step M11) so as to judge whether the activation/execution of stored procedure 13 is necessary or not if virtual table A 131 defined and generated by stored procedure 13 is specified, and it activates stored procedure 13 if it decide that the activation/execution of stored procedure 13 is necessary (Steps M12-M14). If the specified stored procedure 13 was already activated and executed, and RDB 11 has not been renewed since then, stored procedure 13 is not activated. In this case, the processing results of stored procedure 13 already executed are used.

[0047]

Activated stored procedure 13 executes the following processing.

(1) It selects the lines showing the latest time stamps for those with the same merchandise numbers for the respective merchandise numbers in base table B 112 (Step S11).

(2) It correlates the merchandise numbers in base table A 111 with the merchandise numbers obtained as a result of (1) above (Step S12).

(3) It generates a data set comprising the merchandise numbers in base table A 111 and the time stamps obtained as a result of (1) above for virtual table A 131 (Step S13). Resulting output of stored procedure 13 is shown in Figure 4 (c).

[0048]

Then, RDBMS 12 carries out processing based on virtual table B 211 (Step M15). Result of the processing by RDBMS 12 are shown in Figure 4 (d).

/6

[0049]

Subsequently, RDBMS 12 returns the processing results to application program 21 (Step M16).

[0050]

Upon receiving the processing results from RDBMS 12, application program 21 carries out the application processing (Step A13).

[0051]

As described above, application program 21 can obtain the latest data in RDB 11 it needs quickly using virtual table A 131 as the results outputted from stored procedure 13 in place of indexes.

[0052]

Next, a second embodiment of the present invention will be explained in detail in reference to figures. The second embodiment pertains to a system which runs a search in a RDB used to manage 3 base tables pertaining to product problem contents in order to obtain the latest product problem contents for further processing.

[0053]

Figure 6 is a diagram illustrating the configuration of the second embodiment.

[0054]

In reference to Figure 6, in the second embodiment, server 1 and clients 2 are connected via network 3, server 1 is equipped with RDBMS 16 containing stored procedure 17 and stored procedure activation means 14 and connected to RDB 15, and clients 2 are provided with application program 22; wherein, RDB 11, RDBMS 12, stored procedure 13, and application program 21 are replaced by RDB 15, RDBMS 16, stored procedure 17, and application program 22.

[0055]

Here, RDB 15, RDBMS 16, stored procedure 17, and application program 22 which are different from those of the first embodiment will be explained.

[0056]

RDB 15 is a relational database which stores and accumulates data, and it comprises base table A 151, base table B 152, and base table C 153. Base table A 151 is for holding basic data, and base table B 152 and base table C 153 are keeping the history of the basic data.

[0057]

Base table A 151 is one of the tables which constitute RDB 15. Configuration of base table A 151 is shown in Figure 7 (a). Base table A 151 comprises product numbers and product names, and the respective lines are distinguished from each other uniquely using the product numbers as the primary keys.

[0058]

Base table B 152 is one of the tables which constitute RDB 15. Configuration of base table B 152 is shown in Figure 7 (b). Base table B 152 comprises product numbers, branch numbers A, reception numbers, and deletions. The product numbers correspond to the product

numbers in base table A 151, and base table B 152 and base table A 151 are correlated with each other as a result. Branch numbers A indicate the order those with the same product numbers are received, and serial numbers or time stamps may be used to this end also. "Product number + branch number A" is used to distinguish the respective lines uniquely from each other. The reception numbers are serial numbers assigned throughout the products; and every time a product problem is received, a reception number is registered. Content of the problem is registered to base table C 153. If a new problem is received of the same product, its branch number A is incremented, and a new reception number is registered. Deletion indicates whether or not a reception number has been deleted after it was once received, and "DEL" is shown when the reception has been deleted.

[0059]

Base table C 153 is one of the tables which constitute RDB 15. Configuration of base table C 153 is shown in Figure 7 (c). Base table C 153 comprises product numbers, branch numbers A, branch numbers B, and problem contents. "Product number + branch number A" corresponds to "product number + branch number A" in base table B 152, and base table C 153 and base table B 152 are correlated with each other as a result. Branch number B indicates the order of the problem contents within "product number + branch A," and serial numbers or time stamps may be used to this end also. The respective lines are distinguished from each other uniquely using "product number + branch number A + branch number B." Problem contents corresponding to the reception numbers are registered as the problem contents. To renew the problem content pertaining to a reception number which has been received, branch number B is incremented when registering the renewed problem content in order to treat it as a different piece of data during the history management.

[0060]

RDBMS 16 manages RDB 15 and gains access to it. It processes a request issued from application program 22 and returns the processing results corresponding to the request issued to application program 22.

[0061]

Stored procedure 17 defines and generates virtual table A 171 based on base table A 151, base table B 152, and base table C 153 in order to select and generate a dataset. Stored procedure 17 is activated by stored procedure activation means 14. Name of virtual table A 171 is decided when stored procedure 17 is registered to RDBMS 16. Application program 22 can make reference to virtual table A 171 generated by stored procedure 17 by specifying said name.

Furthermore, a well-known method is used as the method for registering stored procedure 17 to RDBMS 16.

[0062]

Virtual table A 171 is a virtual table comprising unique keys with the latest history. It is defined and generated by stored procedure 17 based on base table A 151, base table B 152, and base table C 153; wherein, selection conditions for combining the highest branch number A within the products with the same product numbers in base table A 151 and base table B 152 with the highest branch number B within the same “product number + branch number A” in base table C 153 are specified. Configuration of virtual table A 171 is shown in Figure 7 (c) [sic]. Virtual table A 171 comprises product numbers, branch numbers A, and branch numbers B; wherein, the product numbers, branch numbers A, and branch numbers B correspond to the product numbers in base table A 151, the highest branch numbers A within the products with the same product number in base table B 152, and the highest branch numbers B within the same “product number + branch number A” in base table C 153, respectively. From the viewpoint of application program 22, virtual table A 171 plays the role of indexes which constantly reflect the latest condition of RDB 15 when a reference is made from application program 22.

[0063]

Application program 22 defines virtual table B 221 correlated with virtual table A 171 defined and generated by stored procedure 17 and issues an access request to RDBMS 16 in order to obtain the latest data from RDB 15. Processing results corresponding to the request issued are returned from RDBMS 16, and data processing is carried out based on the data obtained.

[0064]

Virtual table B 221 is a virtual table defined and generated by application program 22 based on virtual table A 171, base table A 151, base table B 152, and base table C 153, and it comprises the data needed by application program 22. Configuration of virtual table B 221 is shown in Figure 7 (d) [sic]. Virtual table B 221 comprises product numbers, product names, reception numbers, and problem contents; wherein, the product numbers, the product names, the reception numbers, and the problem contents correspond to the product numbers in base table A 151 which correspond to the product numbers in virtual table A 171, the product names in base table A 151 which correspond to the product numbers in virtual table A 171, the reception numbers in base table B 152 which correspond to “product number + branch number A” in

virtual table A 171, and the problem contents in base table C 153 which correspond to “product number + branch number A + branch number B” in virtual table A 171, respectively.

[0065]

Operations of the second embodiment of the present invention will be explained in detail in reference to Figure 6 through Figure 9.

[0066]

Figure 8 shows tables for illustrating the operations of the second embodiment; wherein, (a) shows a specific example of base table A 151, (b) shows a specific example of base table B 152, (c) shows a specific example of base table C 153, (d) shows a specific example of virtual table A 171, and (e) shows a specific example of virtual table B 221. In (b), reception numbers “002” and “003” have been deleted after they were once received, and “DEL” is shown under Deletion accordingly. Figure 9 is a flowchart illustrating the flow of the second embodiment.

[0067]

In reference to Figure 9, application program 22 correlates the product numbers in base table A 151 with the product numbers in virtual table A 171, “product number + branch number A” in base table B 152 with “product number + branch number A” in virtual table A 171, and “product number + branch number A + branch number B” in base table C 153 with “product number + branch number A + branch number B” in virtual table A 171, to define virtual table B 221 comprising the product numbers, the product names, the reception numbers, and the problem contents and requests access to RDB 15 to RDBMS 16 (Steps A21-A22).

[0068]

RDBMS 16 analyzes the RDB 15 access request from application program 22 (Step M21) so as to judge whether the activation/execution of stored procedure 17 is necessary or not if virtual table A 171 defined and generated by stored procedure 17 is specified, and it activates stored procedure 17 if it decides that the activation/execution of stored procedure 17 is necessary (Steps M22-M24). If the specified stored procedure 17 was already activated and executed, and RDB 15 has not been renewed since then, stored procedure 17 is not activated. In this case, the processing results of stored procedure 17 already executed are used.

[0069]

Activated stored procedure 17 executes the following processing.

(1) It selects the lines containing the highest branch numbers B within the same “product number + branch number A” for respective “product number + branch number A” (Step S21). In this example, the lines indicated by O marks to the right in Figure 8 (c) are selected.

(2) It correlates “product number + branch number A” in base table B 152 with “product number + branch number A” obtained as a result of (1) above (Step S22). At this time, “product number + branch number A” in base table B 152 for which “DEL” is set under Deletion in base table B 152 are excluded. In this example, because “DEL” is set for the lines for reception numbers “002” and “003,” they are excluded; and the lines attached with © to the right in Figure 8 (b) and (c) are correlated.

(3) It generates a dataset comprising the product numbers in base table A 151, branch numbers A obtained as a result of (2) above, and branch numbers B obtained as a result of (2) above for virtual table A 171 (Step S23). Resulting output of stored procedure 17 is shown in Figure 8 (d).

[0070]

Then, RDBMS 16 carries out processing based on virtual table B 221 (Step M25). Result of the processing by RDBMS 16 are shown in Figure 8 (e).

[0071]

Subsequently, RDBMS 16 returns the processing results to application program 22 (Step M26).

[0072]

Upon receiving the processing results from RDBMS 16, application program 22 carries out the application processing (Step A23).

[0073]

As described above, application program 22 can obtain the latest data in RDB 15 it needs quickly using virtual table A 171 as the results outputted from stored procedure 17 in place of indexes. That is, application program 22 can obtain the latest reception numbers and problem contents pertaining to the product numbers for processing without carrying out complicated processing.

[0074]

In the aforementioned embodiments in accordance with the present invention, the program for executing the processing operations of the database access system is stored as data in a storage device (not illustrated), such as a magnetic disk or an optical disk, and the stored data are read to operate the database access system. When the data for operating the database access system of the present invention are stored in a storage medium, functions of the database access system can be realized as said storage medium is installed.

[0075]

Effects of the invention

A first effect is that the application program can make reference to the latest data at all times simply by making reference to the necessary stored procedure. The reason is that a means for configuring the function to generate indexes for the application program is provided within the stored procedure.

/8

[0076]

A second effect is that burden on the clients can be reduced, and the network traffic is reduced at the same time. The reason is that because the stored procedure is executed by the server's side, there is no need for the application program to issue any complicated SQL, the processing carried out by the application program can be configured simply by describing data operation processing based on a table of unique keys, and the application program at the client's side and the RDBMS at the server's side need to communicate with each other less frequently.

[0077]

A third effect is that in the event of a database problem, no special restoration measure, such as regeneration of a physical index table to provide indexes for the application program, is needed. The reason is that a stored procedure in the form of virtual tables is utilized as indexes for the application program.

Brief description of the figures

Figure 1 is a diagram for illustrating the present invention.

Figure 2 is a diagram for illustrating the configuration of a first embodiment.

Figure 3 are diagrams showing the configurations of (a) base table A, (b) base table B, (c) virtual table A, and (d) virtual table B.

Figure 4 are tables for explaining the operations of the first embodiment; namely, (a) base table A, (b) base table B, (c) virtual table A, and (d) virtual table B.

Figure 5 is a flowchart of the operation flow of the first embodiment.

Figure 6 is a diagram for illustrating the configuration of a second embodiment.

Figure 7 are diagrams showing the configurations of (a) base table A, (b) base table B, (c) base table C, (d) virtual table A, and (e) virtual table B.

Figure 8 are tables for explaining the operations of the second embodiment; namely, (a) base table A, (b) base table B, (c) base table C, (d) virtual table A, and (e) virtual table B.

Figure 9 is a flowchart of the operation flow of the second embodiment.

Figure 10 are tables for explaining the conventional operations; namely, (a) base table A, (b) base table B, and (c) result.

Explanation of the symbols

1	Server
2	Client
3	Network
4	Database
5	Database management system
6	Stored procedure execution means
7	Database access request means
11	RDB
12	RDBMS
13	Stored procedure
14	Stored procedure activation means
15	RDB
16	RDBMS
17	Stored procedure
21	Application program
22	Application program
111	Base table A
112	Base table B
131	Virtual table A
151	Base table A
152	Base table B
153	Base table C
171	Virtual table A
211	Virtual table B
221	Virtual table B

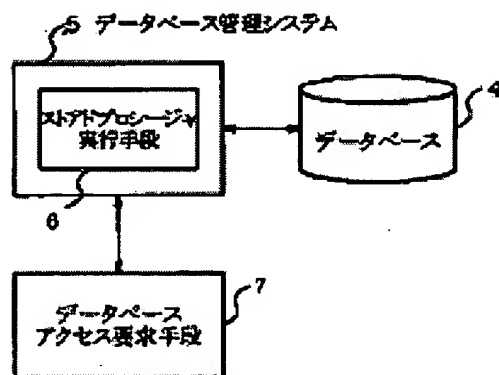


Figure 1

- Key:
- 4 Database
 - 5 Database management system
 - 6 Stored procedure execution means
 - 7 Database access request means

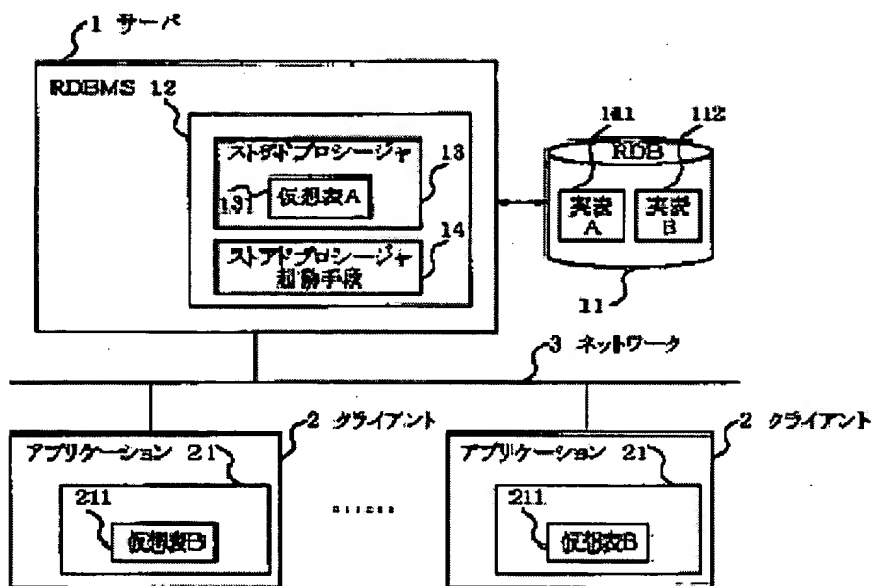


Figure 2

- Key:
- 1 Server
 - 2 Client
 - 3 Network
 - 13 Stored procedure
 - 14 Stored procedure activation means
 - 21 Application program
 - 111 Base table A
 - 112 Base table B

131 Virtual table A
211 Virtual table B

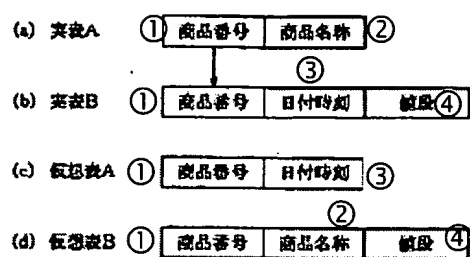


Figure 3

Key: (a) Base table A
(b) Base table B
(c) Virtual table A
(d) Virtual table B
1 Merchandise number
2 Merchandise name
3 Time stamp
4 Price

①	②		①	⑧	⑨
商品番号	商品名称		商品番号	日付時刻	値段
1	あああ	③	1	05281000	200
2	いはい	④	1	05281200	198
3	ううう	⑤	1	05281400	195
4	えええ	⑥	1	05281600	193
5	おおお	⑦	1	06281800	190
			2	05281000	100
			2	06281800	95
			3	05281000	498
			4	06281000	300
			4	06281200	295
			4	05281400	290
			6	05281000	98
			5	05281800	70

(a) 実表A

(b) 実表B

①	②		①	⑧	⑨
商品番号	日付時刻		商品番号	商品名称	値段
1	05281800		1	あああ ③	190
2	05281800		2	いはい ④	95
3	05281000		3	ううう ⑤	498
4	05281400		4	えええ ⑥	290
5	05281800		5	おおお ⑦	70

(c) 仮想表A

(d) 仮想表B

Figure 4

- Key:
- (a) Base table A
 - (b) Base table B
 - (c) Virtual table A
 - (d) Virtual table B
 - 1 Merchandise number
 - 2 Merchandise name
 - 3 A A A
 - 4 I I I
 - 5 U U U
 - 6 E E E
 - 7 O O O
 - 8 Time stamp
 - 9 Price

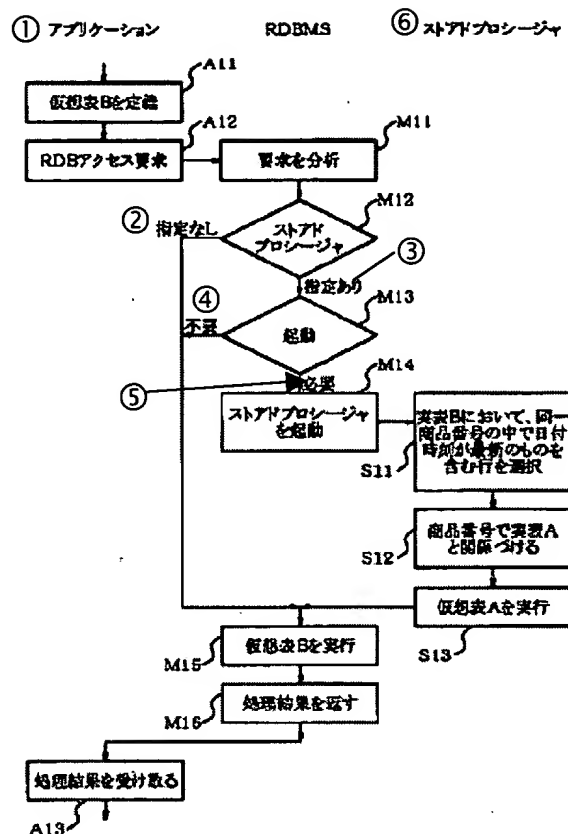


Figure 5

- Keys:
- 1 Application program
 - 2 Not specification
 - 3 Specified
 - 4 Not needed
 - 5 Needed
 - 6 Stored procedure
 - A11 Define virtual table B
 - A12 RDB access request
 - A13 Receive processing results
 - M11 Analyze request
 - M12 Stored procedure
 - M13 Activation
 - M14 Activate stored procedure
 - M15 Execute virtual table B
 - M16 Return processing results
 - S11 Select the lines containing the latest time stamps among those with the same merchandise numbers in base table B
 - S12 Correlate with base table A using merchandise numbers
 - S13 Execute virtual table A

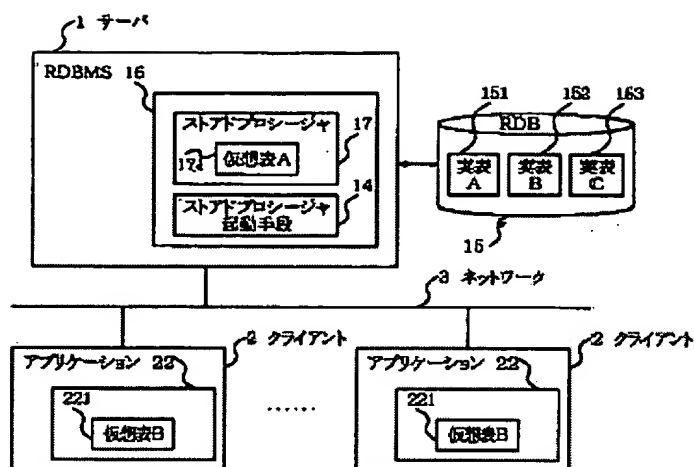


Figure 6

- Key:
- 1 Server
 - 2 Client
 - 3 Network
 - 14 Stored procedure activation means
 - 17 Stored procedure
 - 22 Application program
 - 151 Base table A
 - 152 Base table B
 - 153 Base table C
 - 171 Virtual table A
 - 221 Virtual table B

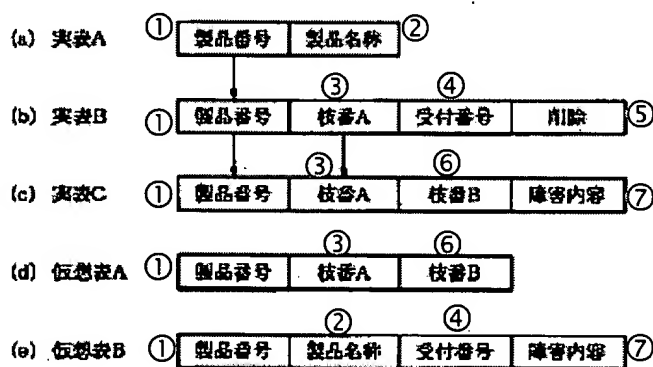


Figure 7

- Key:
- (a) Base table A
 - (b) Base table B
 - (c) Base table C
 - (d) Virtual table A
 - (e) Virtual table B
 - 1 Product number

- 2 Product name
 3 Branch number A
 4 Reception number
 5 Deletion
 6 Branch number B
 7 Problem content

①	②	①	⑥	⑦	④
製品番号	製品名称	製品番号	枝番A	受付番号	削除
1	あい	1	1	001	
2	かき	2	1	002	DEL
3	さし	2	2	004	
		3	1	003	DEL
		3	2	005	

(a) 実表A

①	⑥	⑨	⑩
製品番号	枝番A	枝番B	障害内容
1	1	1	A
1	1	2	B
2	1	1	C
2	2	1	D
3	1	1	E
3	2	1	F
3	2	2	G
3	2	3	H

(b) 実表B

①	⑥	⑨	①	②	⑦	⑩
製品番号	枝番A	枝番B	製品番号	製品名称	受付番号	障害内容
1	1	2	1	あい	001	B
2	2	1	2	かき	004	D
3	2	3	3	さし	005	H

(d) 仮想表A

(e) 仮想表B

Figure 8

- Key: (a) Base table A
 (b) Base table B
 (c) Base table C
 (d) Virtual table A
 (e) Virtual table B
 1 Product number
 2 Product name
 3 A I U
 4 Ka Ki Ku
 5 Sa shi Su
 6 Branch number A

- 7 Reception number
8 Deletion
9 Branch number B
10 Problem content

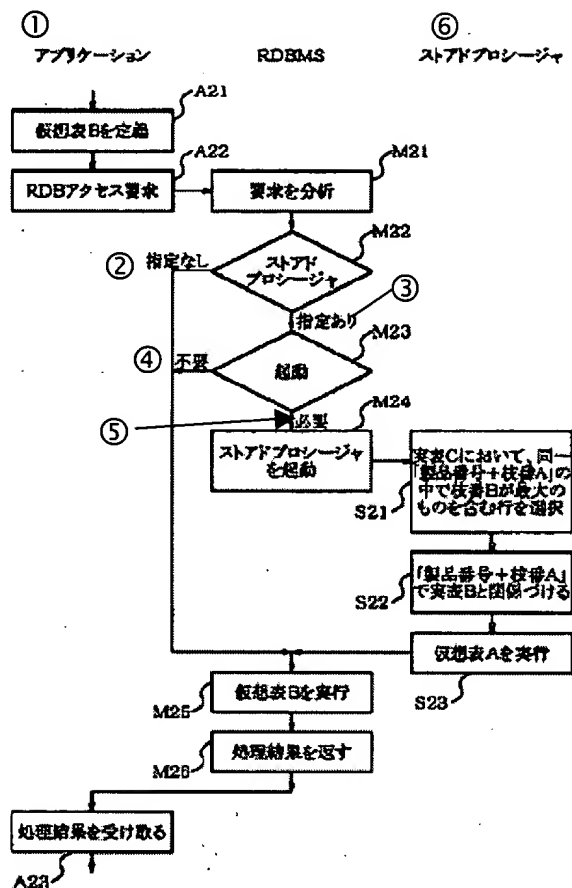


Figure 9

- Key:
- 1 Application program
 - 2 Not specification
 - 3 Specified
 - 4 Not needed
 - 5 Needed
 - 6 Stored procedure
 - A21 Define virtual table B
 - A22 RDB access request
 - A23 Receive processing result
 - M21 Analyze request
 - M22 Stored procedure
 - M23 Activation
 - M24 Activate stored procedure
 - M25 Execute virtual table B
 - M26 Return processing result

- S21 Select the lines containing the latest time stamps among those with the same
 “merchandise number + branch number A” in base table C
 S22 Correlate with base table B using “merchandise number + branch number A”
 S23 Execute virtual table A

①	②	①	⑧	⑨
商品番号	商品名称	商品番号	日付時刻	値段
1	あああ③	1	05261000	200
2	いはい④	1	05261200	198
3	ううう⑤	1	05261400	195
4	えええ⑥	1	05261600	193
5	おとお⑦	1	06261800	190
		2	05261000	100
		2	05261600	95
		3	05261000	498
		4	06261000	300
		4	05261200	295
		4	05261400	290
		5	05261000	98
		5	06261800	70

(a) 実表A

(b) 実表B

①	②	⑨
商品番号	商品名称	値段
1	あああ③	190
2	いはい④	95
3	ううう⑤	498
4	えええ⑥	290
5	おとお⑦	70

(c) 結果

Figure 10

- Key: (a) Base table A
 (b) Base table B
 (c) Result
 1 Merchandise number
 2 Merchandise name
 3 A A A
 4 I I I
 5 U U U
 6 E E E
 7 O O O
 8 Time stamp
 9 Price

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.